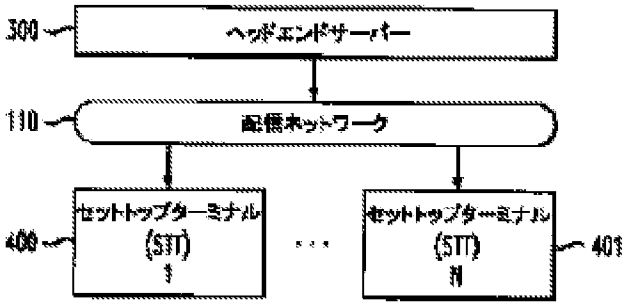
METHOD FOR TRANSMITTING PROGRAM TO LIMIT ACCESS TO END USER AND METHOD FOR DECODING ENCRYPTED PROGRAM

Publication number: JP2001036517
Publication date: 2001-02-09
Inventor: BLEICHENBACHER DANIEL; WOOL AVISHAI
Applicant: LUCENT TECHNOLOGIES INC
Classification:
- international: ***H04N5/44; G09C1/00; H04L9/08; H04N7/08; H04N7/081; H04N7/16; H04N7/167;*** **H04N5/44; G09C1/00; H04L9/08; H04N7/08; H04N7/081; H04N7/16; H04N7/167;** (IPC1-7): H04L9/08; G09C1/00; H04N5/44; H04N7/08; H04N7/081; H04N7/16; H04N7/167
- European: H04H60/149; H04N7/16E2; H04N7/167D
Application number: JP20000135069 20000508
Priority number(s): US19990307643 19990507

Also published as:
EP1051036 (A2)
US6735313 (B1)
EP1051036 (A3)
CA2307157 (A1)

Abstract of **JP2001036517**

PROBLEM TO BE SOLVED: To provide a system to limit access to contents of transmission program such as television program. SOLUTION: A transmitter or a head end server is used by a service provider to transmit encrypted programming contents to one or a plurality of customers. A program identifier (p) used to identify a program is transmitted to the customers together with programming contents. Each customer uses a set-top terminal or an interpretation key to provide a limited access to transmission multimedia information as other device. The set-top terminal 400 or the like receives entitlement information corresponding to a package of one or a plurality of programs that can normally be received for a period from a head end.



(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号
特開2001－36517
（P2001－36517A）
(43)公開日　平成13年２月９日(2001.2.9)

(51)Int.Cl.[7]		識別記号	FI		テーマコード（参考）
Ｈ０４Ｌ	9/08		Ｈ０４Ｌ	9/00	６０１Ｄ
Ｇ０９Ｃ	1/00	６５０	Ｃ０９Ｃ	1/00	６５０Ｚ
Ｈ０４Ｎ	5/44		Ｈ０４Ｎ	5/44	Ａ
	7/08			7/16	Ｃ
	7/081		Ｈ０４Ｌ	9/00	６０１Ｅ

審査請求　未請求　請求項の数29　ＯＬ　外国語出願　（全 46 頁）　最終頁に続く

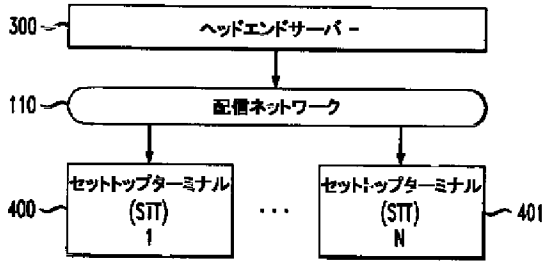
(21)出願番号　特願2000－135069(P2000－135069)
(22)出願日　平成12年５月８日(2000.5.8)
(31)優先権主張番号　０９／３０７６４３
(32)優先日　平成11年５月７日(1999.5.7)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　59607/259
ルーセント　テクノロジーズ　インコーポレイテッド
Ｌｕｃｅｎｔ　Ｔｅｃｈｎｏｌｏｇｉｅｓ　Ｉｎｃ．
アメリカ合衆国　07974　ニュージャージー、マレーヒル、マウンテン　アベニュー　600－700
(74)代理人　100081053
弁理士　三俣　弘文

最終頁に続く

(54)【発明の名称】　エンドユーザに対してアクセス制限することができるプログラムを送信する方法、暗号化されたプログラムをデコードする方法

(57)【要約】
【課題】　テレビジョンなどの送信プログラミング内容へのアクセス制限するシステムを提供する。
【解決手段】　送信器ないしヘッドエンドサーバを用いてサービスプロバイダーによって１もしくは複数の顧客に暗号化されたプログラミング内容が送信される。プログラムを識別するのに用いるプログラム識別子ｐは、プログラミング内容と共に顧客に送信される。各顧客は、セットトップターミナルないし解読キーを用いて送信マルチメディア情報に制限されたアクセスを与える他の機構を備える。セットトップターミナルは顧客がある期間に正規に受信できる１もしくは複数のプログラムのパッケージに対応するエンタイトルメント情報をヘッドエンドから受信する。



【特許請求の範囲】
【請求項１】　エンドユーザに対してアクセス制限することができるプログラムを送信する方法であって、
（Ａ）バイナリ値を有するプログラム識別子を前記プログラムに割り当てるステップと、
（Ｂ）少なくとも１つのマスターキーを定めるステップと、
（Ｃ）前記プログラム識別子のバイナリ値に基づいて前記マスターキーに少なくとも１つのハッシュ関数を適用することにより得たプログラムキーを用いることにより前記プログラムを暗号化するステップと、
（Ｄ）前記暗号化したプログラムを前記プログラム識別子とともに前記エンドユーザへと送るステップとを有することを特徴とする方法。
【請求項２】　前記プログラム識別子はｎビットからなり、
前記プログラム識別子の対応するビット値に従って、前記プログラム識別子のｎビットの位置それぞれに前記ハッシュ関数の１つが適用されることを特徴とする請求項１記載の方法。
【請求項３】　（Ｅ）前記エンドユーザにより得たプログラムのセットに基づいて前記エンドユーザにエンタイトルメント情報を提供するステップをさらに有することを特徴とする請求項１記載の方法。
【請求項４】　前記エンタイトルメント情報には、前記エンドユーザにより得たプログラムのセットに基づくキーツリーの一部を含むことを特徴とする請求項３記載の方法。
【請求項５】　前記エンドユーザは、記憶された前記エンタイトルメント情報から前記プログラムキーを得るために前記プログラム識別子を用いることを特徴とする請求項３記載の方法。
【請求項６】　前記プログラム識別子は前記暗号化プログラムの送信とともにインターリーブされることを特徴とする請求項１記載の方法。
【請求項７】　前記プログラム識別子は、制御チャネル上で送信されることを特徴とする請求項１記載の方法。
【請求項８】　複数のエンドユーザにプログラムを送信する方法であって、
（Ａ）プログラム識別子を有するプログラムを、前記プログラム識別子のビット位置それぞれのバイナリ値に基づくマスターキーにハッシュ関数を回帰的に適用することによって得たプログラムキーを用いて暗号化するステップと、
（Ｂ）暗号化したプログラムおよび前記プログラム識別子を前記エンドユーザに送信するステップとを有することを特徴とする方法。
【請求項９】　前記プログラム識別子はｎビットからなり、
前記プログラム識別子の対応するビット値に従って、前記プログラム識別子のｎビットの位置それぞれに前記ハッシュ関数が適用されることを特徴とする請求項８記載の方法。
【請求項１０】　（Ｃ）前記エンドユーザにより得たプログラムのセットに基づいて前記エンドユーザにエンタイトルメント情報を提供するステップをさらに有することを特徴とする請求項８記載の方法。
【請求項１１】　前記エンタイトルメント情報には、前記エンドユーザにより得たプログラムのセットに基づくキーツリーの一部を含むことを特徴とする請求項１０記載の方法。
【請求項１２】　前記エンドユーザは、記憶された前記エンタイトルメント情報から前記プログラムキーを得るために前記プログラム識別子を用いることを特徴とする請求項１０記載の方法。
【請求項１３】　前記プログラム識別子は前記暗号化プログラムの送信とともにインターリーブされることを特徴とする請求項８記載の方法。
【請求項１４】　前記プログラム識別子は、制御チャネル上で送信されることを特徴とする請求項８記載の方法。
【請求項１５】　少なくとも１つのプログラムパッケージに対応するプログラムを複数のエンドユーザに送信する方法であって、
（Ａ）前記エンドユーザにより得たプログラムのセットに基づいて前記エンドユーザにエンタイトルメント情報を提供するステップと、
（Ｂ）プログラム識別子を有するプログラムを、前記プログラム識別子のビット位置それぞれのバイナリ値に基づくマスターキーにハッシュ関数を回帰的に適用することによって得たプログラムキーを用いて暗号化するステップと、
（Ｃ）暗号化されたプログラムとともに前記プログラム識別子を前記エンドユーザに送信するステップとをさらに有し、
前記エンドユーザが前記プログラムの正当ユーザであれば、前記エンドユーザは記憶されたエンタイトルメント情報から前記プログラムキーを得ることを特徴とする方法。
【請求項１６】　前記プログラム識別子はｎビットからなり、
前記プログラム識別子の対応するビット値に従って、前記プログラム識別子のｎビットの位置それぞれに前記ハッシュ関数の１つが適用されることを特徴とする請求項１５記載の方法。
【請求項１７】　前記エンタイトルメント情報には、前記エンドユーザにより得たプログラムのセットに基づくキーツリーの一部を含むことを特徴とする請求項１５記載の方法。
【請求項１８】　前記エンドユーザは、記憶された前記

エンタイトルメント情報から前記プログラムキーを得るために前記プログラム識別子を用いることを特徴とする請求項１５記載の方法。

【請求項１９】　前記プログラム識別子は前記暗号化プログラムの送信とともにインターリーブされることを特徴とする請求項１５記載の方法。

【請求項２０】　前記プログラム識別子は、制御チャネル上で送信されることを特徴とする請求項１５記載の方法。

【請求項２１】　暗号化されたプログラムをデコードする方法であって、

（Ａ）前記プログラムのプロバイダーから前記顧客が得たプログラムのセットに基づいてキーツリーから少なくとも１つの中間キーを含むエンタイトルメント情報を受信するステップと、

（Ｂ）プログラムキーで暗号化された暗号化プログラムとプログラム識別子を受信するステップと、

（Ｃ）前記プログラム識別子及び前記キーツリーの記憶された部分から前記プログラムキーを得るステップと、

（Ｄ）前記プログラムキーを用いて前記暗号化プログラムを解読するステップとを有することを特徴とする方法。

【請求項２２】　前記プログラム識別子はｎビットからなり、

前記マスターキーは前記キーツリーのルートに配置され、前記キーツリーは、ｎのツリーレベルが作られるまで各ノードにハッシュ関数を適用することにより前記キーツリーが生成されることを特徴とする請求項２１記載の方法。

【請求項２３】　暗号化されたプログラムをデコードする方法であって、

（Ａ）前記プログラムのプロバイダーから、顧客が得るプログラムのセットに基づくキーツリーから少なくとも１つの中間キーを含むエンタイトルメント情報を受信するステップと、

（Ｂ）プログラムキーで暗号化された暗号化プログラムとプログラム識別子を受信するステップと、

（Ｃ）前記プログラム識別子のバイナリ値に基づいて前記中間キーにハッシュ関数を回帰的に適用することにより前記プログラム識別子及び前記中間キーからキーツリーの記憶された部分から前記プログラムキーを得るステップと、

（Ｄ）前記プログラムキーを用いて前記暗号化プログラムを解読するステップとを有することを特徴とする方法。

【請求項２４】　前記プログラム識別子はｎビットからなり、

前記中間キーは前記キーツリーのレベルｒにおける中間ノードに対応し、前記ハッシュ関数は前記中間キーにｎ－ｒ回適用されることを特徴とする請求項２３記載の方法。

【請求項２５】　エンドユーザへのアクセスを制限するプログラムを送信するシステムであって、

（Ａ）マスターキーとコンピュータ読みとり可能コードを記憶するメモリーと、

（Ｂ）前記メモリーに動作的につながったプロセッサとを有し、このプロセッサが、

（ａ）バイナリ値を有するプログラム識別子を前記プログラムに割り当て、

（ｂ）少なくとも１つのマスターキーを定め、

（ｃ）前記プログラム識別子のバイナリ値に基づいて前記マスターキーに少なくとも１つのハッシュ関数を適用することによりプログラムキーを用いて前記プログラムを暗号化し、

（ｄ）前記プログラム識別子とともに暗号化プログラムを前記エンドユーザに送信するように構成することを特徴とするシステム。

【請求項２６】　エンドユーザーに対するアクセスが制限されたプログラムを送信するシステムであって、

（Ａ）マスターキーおよびコンピュータ読み取り可能コードを記憶するメモリーと、

（Ｂ）前記メモリーに動作上つながったプロセッサーとを有し、

前記プロセッサーは、

（ａ）前記プログラム識別子のビット位置それぞれのバイナリー値に基づいてマスターキーにハッシュ関数を回帰的に適用することによって得られるプログラムキーを用いて、プログラム識別子を有する該プログラムを暗号化し、

（ｂ）前記エンドユーザーに暗号化された該プログラムおよび前記プログラム識別子を送信するように構成することを特徴とするシステム。

【請求項２７】　暗号化されたプログラムをデコードするシステムであって、

（Ａ）マスターキーおよびコンピュータ読み取り可能コードを記憶するメモリーと、

（Ｂ）前記メモリーに動作上つながったプロセッサーとを有し、前記プロセッサーは、

（ａ）前記顧客によって得られるプログラムのセットに基づくキーツリーの部分を含むエンタイトルメント情報を該プログラムのプロバイダーから受信し、

（ｂ）プログラムキーによって暗号化された暗号化プログラム、およびプログラム識別子を受信し、

（ｃ）前記プログラム識別子および前記キーツリーの記憶された前記部分から前記プログラムキーを得て、

（ｄ）前記プログラムキーを用いて前記暗号化プログラムを解読するように構成することを特徴とするシステム。

【請求項２８】　コンピューター読み取り可能コード手段が実装されたコンピュータ読み取り可能媒体であっ

て、該コンピュータ読み取り可能手段は動作時に、
（ａ）バイナリー値を有するプログラム識別子をプログラムに割り当て、
（ｂ）少なくても一つのマスターキーを定め、
（ｃ）前記プログラム識別子のバイナリー値に基づいて前記マスターキーに少なくとも一つのハッシュ関数を適用することによって得られるプログラムキーを用いて、該プログラムを暗号化し、
（ｄ）前記プログラム識別子とともに暗号化された該プログラムをエンドユーザーに送信することを特徴とするコンピュータ読み取り可能媒体。
【請求項２９】　コンピューター読み取り可能コード手段が実装されたコンピュータ読み取り可能媒体であって、該コンピュータ読み取り可能手段は動作時に、
（ａ）前記顧客によって得られるプログラムのセットに基づくキーツリーの部分を含むエンタイトルメント情報を該プログラムのプロバイダーから受信し、（ｂ）プログラムキーによって暗号化された暗号化プログラム、およびプログラム識別子を受信し、
（ｃ）前記プログラム識別子および前記キーツリーの記憶された前記部分から前記プログラムキーを得て、
（ｄ）前記プログラムキーを用いて前記暗号化プログラムを解読することを特徴とするコンピュータ読み取り可能媒体。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、送信プログラミング内容へのアクセスを制限するシステムに関し、特に、プログラムを解読するのに必要な解読キーを得るために、記憶されたエンタイトルメント情報と共に、セットトップターミナルによって用いられるプログラム識別子を用いて解読されたプログラムを送信するシステムに関する。
【０００２】
【従来の技術】テレビジョン視聴者が利用可能なチャンネルの数が増え、そのようなチャネルで利用可能なプログラミング内容の範囲が増えるに従い、テレビジョン視聴者の人口の多数を満足させるチャンネルやプログラムのパッケージをケーブルテレビジョンオペレーターやデジタル衛星サービスオペレーターのようなサービスプロバイダーが提供することはますます重要になっている。顧客に提供されるパッケージの開発は一般にマーケティング機能である。一般にサービスプロバイダーは様々なサイズのパッケージを提供することを望む。例えば、一つのプログラムから全てのプログラム、それらの間の組み合わせなどである。
【０００３】サービスプロバイダーは通常、「ヘッドエンド」と呼ばれる送信器から多数の顧客へとテレビジョンプログラムをブロードキャストする。各顧客は受けるプログラミングの一部のみに通常関わる。例えば、無線放送環境において、送信されるプログラミングはアンテナや衛星円盤のような適切な受信器によって何れの人でも受信することができる。パッケージを購買した正規顧客のみにプログラムのアクセスを制限するために、サービスプロバイダーは通常送信プログラムを暗号化し、顧客に１もしくは複数の暗号器を含む。セットトップターミナル（ＳＴＴ）を提供する。このような方法で、セットトップターミナルは暗号化送信を受信し、顧客が見るプログラムを暗号化する。これ以外はなにもしない。
【０００４】セットトップターミナルに記憶された機密性が高い情報の海賊行為を最小にするため、セットトップターミナルは通常セキュアプロセッサーやセキュアメモリーを備える。このセキュアメモリーは、数キロビットのオーダーのキャパシティを有し暗号キーを記憶する。セキュアメモリーは一般に揮発性ではなくタンパーレジスタントである。また、セキュアメモリーは書き込み可能であることが多く、各課金周期毎にキーをリプログラムすることができる。従来のセットトップターミナルのセキュアメモリーキャパシティが制限されているので、記憶されるキーの数を制限してしまい、サービスプロバイダーが提供するパッケージの数も制限してしまう。月単位の課金周期にサービスプロバイダーがブロードキャストするプログラムの数は通常、２０万のオーダーであることがある。
【０００５】従来のセットトップターミナルは、サービスプロバイダーが提供するプログラムの各パッケージに対応するビットエントリーを有するビットヴェクトルを含むものがある。もし特定の顧客がパッケージの正規受信者であれば、セットトップターミナルに記憶されるビットベクトルにおけるビットエントリーは「１」にセットされる。その後に、サービスプロバイダーが送信する全てのプログラムは一つのキーで暗号化される。プログラムを受けると、セットトップターミナルは、ビットベクトルにアクセスし、対応するビットエントリーがセットされているかどうかを判断する。もしビットエントリーがセットされていれば、セットトップターミナルは一つの記憶された暗号器を用いてプログラムを解読する。
【０００６】理論上は各パッケージ（パッケージは一般に一つのプログラムで構成する）に対し一つのビットエントリーを提供することによりビットベクトル方式にて柔軟性が達成されているように見えるが、ビットベクトルの長さは一つの課金期間に多くのプログラムを送信するシステムにおいて実用的ではない。また、このようなシステムにおけるアクセス制御はビットベクトルにおけるエントリーによって排他的に与えられ、暗号的（cryptographic）ではない。従って、もし顧客がビットベクトルを書き込み、全てのビットを「１」にセットすることができれば、顧客は全てのプログラムにアクセスすることができてしまう。
【０００７】また、プログラムを各パッケージに分け、

パッケージにおける全てのプログラムが同じキーを用いて暗号化されるものがある。各パッケージは一つのテレビジョンチャンネルに対応する。セットトップターミナルはその顧客が正規受信者である各パッケージに対しての解読キーを記憶する。従って、もし複数のパッケージにプログラムを含ませるのであれば、そのプログラムは対応する各パッケージ毎に再送信しなければならず、この各送信において特定のパッケージに対応する暗号キーによって暗号化される。アクセス制御は暗号学的であるが、何回もプログラミングを再送信することに関するオーバーヘッドによって、多数のパッケージに同じプログラムを配置することを現実的ではなくし、プログラムのパッケージの設計において柔軟性を制限してしまう。

【０００８】このようなプログラム内容を暗号化し送信する従来のシステムは、正規顧客のみにアクセスを制限することに関して比較的成功しているが、テレビジョンネットワークのようなサービスプロバイダーがセットトップターミナルの制限されたセキュアメモリーキャパシティを越えずに、また、オーバーヘッドを相当に増加させずに多数のプログラムを含む多数の異なるパッケージを顧客に提供することを可能にしていない。米国特許出願０８／９１２１８６（１９９７年８月１５日出願）に記載された「Vspaceシステム」には、送信プログラミング内容へのアクセスを制限する暗号学的方法および装置が記載されている。

【０００９】Vspaceシステムにおける各プログラムは、プログラムキーk_pを用いて送信の前にヘッドエンドサーバによって暗号化される。プログラムキーそれぞれは、マスターキーMの定められたセットの線形組み合わせである。プログラムを識別するプログラム識別子は、暗号化プログラミング内容と共に送信される。顧客のセットトップターミナルは、受信したプログラム識別子pおよび前に記録したエンタイトルメント情報のみから解読キーを得ることができる。Vspaceシステムは、プログラムヘッダーを相当に拡張せずに（プログラムと共にプログラム識別子のみが送信される）柔軟性のあるパッケージを可能にしながら暗号学的アクセス制御メカニズムを提供する。なぜなら、対応する各パッケージ毎にプログラムを再送信する必要がないからである。

【００１０】

【課題を解決するための手段】一般に、送信器ないしヘッドエンドサーバを用いてサービスプロバイダーによって１もしくは複数の顧客に暗号化されたプログラミング内容が送信される。プログラムを識別するのに用いるプログラム識別子pは、プログラミング内容と共に顧客に送信される。各顧客は、セットトップターミナルないし解読キーを用いて送信マルチメディア情報に制限されたアクセスを与える他の機構を備える。セットトップターミナルは顧客がある期間に正規に受信できる１もしくは複数のプログラムのパッケージに対応するエンタイトルメント情報をヘッドエンドから受信する。

【００１１】各プログラムはプログラムキーk_pを用いて送信の前にヘッドエンドサーバにより暗号化される。個のプログラムキーk_pはそのプログラムにユニークなようにすることができる。暗号化されたプログラムの送信に加えて、ヘッドエンドサーバはセットトップターミナルにプログラム識別子pを送信する。セットトップターミナルは記憶されたエンタイトルメント情報と共に受信したプログラム識別子pを用い、プログラムを解読するのに必要な解読キーを得る。この方法において、もし顧客が特定のプログラムの正規利用者であれば、セットトップターミナルは記憶され受信された情報を用いて暗号化されたプログラムキーk_pを得ることができ、その後でそのプログラムキーk_pを用いて暗号化されたプログラムを解読することができる。実施例において、プログラム識別子pは、プログラムの一部に対しインターリーブされることができ、別々の専用制御チャネル上で送信されることができる。

【００１２】送信プログラムを暗号化するのに用いられるｋ－ビットプログラムキーk_pのそれぞれは、マスターキーmに１もしくは複数の擬似ランダムハッシュ関数を適用することによって得ることができる。例として、長さを２倍にするハッシュ関数Hを用いることができる。従って、ハッシュ関数Hはｋビットバイナリー値を取り、２ｋの長さのバイナリー値を作る。ハッシュ関数Hの出力はｋ－ビットバイナリー値の対H_0とH_1として表すことができる。ここで、H_0は当該ハッシュ関数の出力の左半分として識別することができ、H_1は当該ハッシュ関数の出力の右半分として識別することができる。

【００１３】例として、プログラム識別子pの各ビット位置の対応するバイナリー値に従って、マスターキーに対しハッシュ関数H_0またはH_1を回帰的に適用することによってプログラムキーk_pを得ることができる。従って、もしプログラム識別子pがmビットからなるのであれば、ハッシュ関数H_0またはH_1の一方がプログラム識別子pの対応するビット値に従ってプログラム識別子pのｎのビット位置それぞれに対して適用される。最初には、ハッシュ関数H_0またはH_1の一方がプログラム識別子pの最左桁ビットのバイナリー値に従ってマスターキーに適用される。その後で、残りの（ｎ－１）ビット位置それぞれに対し、対応するビットのバイナリ値に従って、前のハッシュ演算の結果にハッシュ関数H_0またはH_1の一方が適用される。プログラムキーk_pの計算は以下のように表すことができる。

【数１】

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

【００１４】このようなハッシュ演算は、ツリーのルート２マスターキーmが配置されているようなｎレベルバ

イナリーツリーＴ（キーツリーとも呼ばれる）に関連して表すことができる。所望の数のツリーレベル（ｎ）が作られるまで、各ノードに対しハッシュ関数H_0およびH_1を適用することによりツリーを生成することができる。プログラムキーk_pはツリーのボトム（底）レベルにおけるリーフ（葉）ノードに対応する。各プログラムキーk_pに対応するバイナリーインデックス（そして同様にプログラム識別子ｐ）は、ルートから所望のリーフノードへのキーツリーを通るパス（路）に対応する。従って、ノードｕのインデックスないしラベルは、ルートからノードｕへのパス上のＨ上のラベルの連結である。Ｔ（ｕ）はノードｕをルートとするサブツリー、即ち、ノードｕのサブツリーにおけるリーフに対応するプログラム識別子ｐのセットを表す部分的プログラム識別子ｐを有するキーツリーにおける深さｒにおける内部ノードｕ（ｕ１,...,ur）に対して、サブツリーＴ（ｕ）における何れのプログラムのキーをもハッシュ関数を（ｎ－ｒ）回作動させることにより計算することができる。

【００１５】

【発明の実施の形態】図１は、ヘッドエンドサーバー３００のような送信器を用いてサービスプロバイダーから１もしくは複数の配信ネットワーク１１０を介してセットトップターミナル４００～４０１を有する１もしくは複数の顧客へとビデオ、オーディオ、データのような暗号化マルチメディア情報を転送するネットワーク環境を示してある。このヘッドエンドサーバー３００は図３に関連して下で議論し、セットトップターミナル４００は図４に関連して下で議論する。本明細書において、セットトップターミナルは、解読キーを用いて送信されたマルチメディア情報にアクセス制限を与えるいずれの機構をも含む。例えば、コンピュータ構成や通信デバイスを含む。セットトップターミナルが実行するソフトウェアはサービスプロバイダーがダウンロードするものであってもよい。ネットワーク１１０はデジタルサテライトサービス（ＤＳＳ™）のようなプログラミング内容を配信する無線ブロードキャストネットワーク、ケーブルテレビジョンネットワーク（ＣＡＴＶ）、公衆交換ネットワーク（ＰＳＴＮ）、光ネットワーク、ＩＳＤＮ、インターネットのような有線ネットワークとすることができる。

【００１６】セットトップターミナル４００はヘッドエンドサーバー３００からエンタイトルメント情報を間欠的に受信し、ある時間間隔の間（例えば、課金周期）顧客が正規ユーザーであるプログラムに顧客がアクセスすることを可能にする。本明細書において、パッケージは、所定のプログラムのセットであり、あるプログラムは１もしくは複数のパッケージに属することができる。プログラムは、テレビジョンのエピソードや映画のような特定の長さの連続的なマルチメディア送信のいずれをも意味する。エンタイトルメント情報は、いずれの適切なセキュア単方向または双方向プロトコルを用いてヘッドエンドサーバー３００からセットトップターミナル４００にダウンロードすることができる。

【００１７】プログラムキーおよびプログラム識別子_各送信プログラムはプログラムキーk_pを用いてヘッドエンドサーバー３００によって暗号化される。このプログラムキーk_pはプログラムにユニークなものとすることができる。適切な暗号化およびセキュリティー技術に関しては、文献、B. Schneier,Applied Cryptography(2d ed. 1997)に記載されている。暗号化プログラムの送信に加えて、ヘッドエンドサーバー３００はセットトップターミナル４００にｎビットプログラム識別子をも送信する。これは、記憶されたエンタイトル情報とともにセットトップターミナル４００によって用いられ、下で詳細に示すように、プログラムを解読するのに必要な解読キーを得る。

【００１８】プログラムへのプログラム識別子の割り当てと題する下の項目で説明するように、プログラム識別子ｐは任意に選ばれるのではない。好ましい実施例において、プログラム識別子ｐはＭＰＥＧ－２標準に規定されたＥＣＭフィールドにて送信される３２ビット値から成ることができる。この場合、もし顧客が特定のプログラムの正規ユーザーであれば、セットトップターミナル４００は記憶され受信された情報からプログラムキーk_pを得ることができ、その後で暗号化プログラムを解読するようにプログラムキーk_pを用いることができる。

【００１９】本発明の更なる特徴によれば、暗号化送信プログラムに用いられるｋビットのプログラムキーk_pのそれぞれは、マスターキーｍに１もしくは複数の擬似ランダムハッシュ関数を適用することにより得ることができる。適切な擬似ランダムハッシュ関数の説明は、文献、O. Goldreich et al., "How to Construct Random Functions, "J. ACM, 33:792-807(1986)に記載されている。

【００２０】例として、暗号学的にセキュアであり、長さを２倍にするハッシュ関数を以下のように用いる。

$H: \{0,1\}^k \rightarrow \{0,1\}^{2k}$

ここで、ｋはプログラムキーk_pの長さである。従って、ハッシュ関数Ｈはｋビットのバイナリー値を取り、長さ２ｋのバイナリー値を作る。このハッシュ関数Ｈの出力はｋビットバイナリー値の対H_0とH_1として表すことができる。ここで、H_0はハッシュ関数Ｈの出力の左側半分（左側桁ビット）であり、Ｈ　{1}はハッシュ関数Ｈの出力の右側半分（右側桁ビット）である。H_0とH_1は別々のハッシュ関数と呼ぶことができる。

【００２１】ｋ＝１６０であれば、Ｈは、文献、Secure Hash Standard, National Institute of Standards and Technology, NIST FIPS PUB 180-1, U. S. Dept. of Commerce(April, 1995)に記載されるような秘密ハッシュ標準ＳＨＡ－１を用いて規定することができる。即

ち、H_0はＳＨＡ－１（ｘ‖０）となり、H_1はＳＨＡ－１（ｘ‖１）となる。ここで、０と１はそれぞれ全て０のビットストリング、全て１のビットストリングである。

【００２２】プログラムキーk_pは、プログラム識別子ｐのバイナリー値に従ってマスターキーｍに１もしくは複数のハッシュ関数を回帰的に適用することによって得ることができる。例として、プログラムキーk_pは、プログラム識別子ｐの各ビット位置のバイナリー値に従ってマスターキーｍにハッシュ関数H_0またはH_1の一方を回帰的に適用することによって得ることができる。一般に、もしプログラム識別子ｐがｎビットから成れば、プログラム識別子ｐの対応するビット値に従ってプログラム識別子ｐのｎのビット位置のそれぞれにハッシュ関数H_0またはH_1の一方が適用される（最左ビットから開始する）。

【００２３】最初にハッシュ関数H_0またはH_1の一方が最左桁ビットのバイナリー値に従ってマスターキーに適用される。その後で、残りの（ｎ－１）ビット位置それぞれに対し、ハッシュ関数H_0またはH_1の一方が対応するビットのバイナリー値に従って、前のハッシュ操作の結果に適用される。下の「キーツリー」という題の項目で説明するように、このハッシュ操作は以下のように表すことができる。

【数２】

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

【００２４】上述のように、ヘッドエンドサーバー３００は暗号化プログラムとともにプログラム識別子ｐを送信する。従って、プログラム識別子ｐが与えられるとセットトップターミナル４００は受信プログラムの解読に用いられるプログラムキーk_pを得なければならない。上述のように、プログラムキーk_pはプログラム識別子ｐのバイナリー値に従ってマスターキーｍに１もしくは複数のハッシュ関数を回帰的に適用することによって得ることができる。プログラムキーk_pは、下で説明する記憶されたエンタイトルメント情報および受信したプログラム識別子ｐを間接的に用いて顧客のセットトップターミナル４００によって得られなければならない。

【００２５】キーツリー

上で説明したように、プログラムキーk_pは、プログラム識別子ｐのバイナリー値に従ってマスターキーｍに１もしくは複数のハッシュ関数を回帰的に用いることによって得ることができる。単一のｋビットのマスターキーｍを用いる。プログラム識別子ｐのビットはｐ＝（$p_1,\ldots,p_n$）として表すことができる。ここで、p_1は最左桁ビットであり、最右桁ビットである。プログラム識別子ｐを有するプログラムに対する暗号化キーk_pは以下のように定めることができる。

【数３】

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

【００２６】ハッシュ操作は、図２に示したキーツリー２００のような完全なｎレベルバイナリーツリーＴとして表すことができる。図２に示したキーツリー２００は、３ビットからなるプログラム識別子ｐを有する実装例に対応する。図２に示すように、マスターキーｍがツリー２００のルート２１０に配置される。プログラムキーk_pはリーフノード２４０～２４７のようなリーフノードに対応する。デリーフノード２４３のプログラムキーk_pに対応するインデックス０１１のような図２に示す各プログラムキーk_pに対応するインデックスは、ルート２１０からリーフノード２４３へのキーツリー２００を通してのパスを示す。例えば、２４３のプログラムキーk_pは、ルート２１０からの左エッジ（H_0）、ノード２２０からの右エッジ（H_1）、ノード２３２からの右エッジ（H_1）とたどることによって得ることができる。即ち、H_0がH_1が第２のハッシュ結果に更に適用される。プログラムキーk_{p011}を得ることができる。

【００２７】従って、ノード２４３のようなノードｕのラベルは、ルート２１０からノードｕへのパスのエッジ上のラベルを連結したものとなっている。各ノードのラベルはプログラム識別子ｐで特定することができる。ノードｕをルートとするサブツリーを表すために（即ち、ノードｕのサブツリーにおけるリーフに対応するプログラム識別子ｐのセットを表すために）、Ｔ（ｕ）が用いられる。キーツリー２００における深さｒにおける内部ノードｕは、部分的プログラム識別子ｐ（$u_1,\ldots,u_r$）を有し、これらに対し、サブツリーＴ（ｕ）におけるいずれのプログラムのキーを計算することができる。ノードｕのサブツリーにおけるいずれのプログラムのキーをもハッシュ関数を（ｎ－ｒ）回作動させることにより計算することができる。具体的には、適切なハッシュ関数H_0またはH_1をプログラム識別子ｐの（ｎ－ｒ）の低い桁のビットそれぞれの値が指示するように用いる。従って、ノードｕに対応するプログラムキーk_pは、ノードｕのサブツリーにおける全てのプログラムに対するエンタイトルメントとして機能することができる。

【００２８】もし関数Ｈが擬似ランダム発生器であれば、マスターキーｍによりパラメータ化されたプログラムキーのマッピング$k_p\{0,1\}^n \rightarrow \{0,1\}^k$は擬似ランダム関数である。これについては文献、O. Goldreich et al., "How toConstruct Random Functions," J. ACM, 33:792-807(1986)に記載されている。

【００２９】システムコンポーネント

図３はヘッドエンドサーバー３００のアーキティクチャーを示すブロック図である。ヘッドエンドは、テレビジョンネットワーク、ケーブル運用者、デジタル衛星サービス運用者、あるいは暗号化プログラミング内容を送信

する任意のサービスプロバイダーに関連づけられるものとすることができる。ヘッドエンドサーバー３００は例えば、IBM Corp、製造するＲＳ６０００サーバーにて実装することができ、本発明の機能および動作を実行させることができる。ヘッドエンドサーバー３００にはプロセッサー３１０およびデータ記憶デバイス３２０のような関連するメモリーを備える。プロセッサー３１０は単一のプロセッサーとして実装してもよく、並列に動作する幾つかのプロセッサーとして実装してもよい。データ記憶デバイス３２０やＲＯＭに１もしくは複数の命令を記憶させ、プロセッサー３１０が取り出し、解釈し、実行できるようにする。

【００３０】上述のように、データ記憶デバイス３２０はマスターキーｍを記憶するマスターキーデータベース３５０を備える。例えば、マスターキーｍは課金周期毎のように更新することができる。また、下で図５に関連して説明するように、データ記憶デバイス３２０はプログラムデータベース５００を有する。プログラムデータベース５００はプログラム識別子ｐおよび各プログラムに対応する関連するパッケージを提示する。また、図７、８に関連して説明するように、データ記憶デバイス３２０はエンタイトルメント情報配信プロセス７００およびプログラム配信プロセス８００を有する。

【００３１】一般に、エンタイトルメント情報配信プロセス７００は正規ユーザーであるプログラムにアクセスするのに各顧客が必要とするエンタイトルメント情報を生成し配信する。また、プログラム配信プロセス８００は、プログラム識別子ｐでプログラムを暗号化し送信するために、プログラムに割り当てられたプログラム識別子ｐに基づいてプログラムキーk_pを得る。

【００３２】通信ポート３３０はヘッドエンドサーバー３００をネットワーク１１０につなぎ、図１に示したセットトップターミナル４００のようなつながれた受信器それぞれにヘッドエンドサーバー３００をリンクする。

【００３３】図４は、セットトップターミナル４００のアーキティクチャーを示すブロック図である。セットトップターミナル４００は、例えば、テレビジョンに対応するセットトップターミナル（ＳＴＴ）として実装することができ、本発明の機能や動作を行うように変更することができる。セットトップターミナル４００は、プロセッサー４１０およびデータ記憶装置４２０のようなメモリー、通信ポート４３０を備え、図３に関連した上のようなハードウェアと同様な方法で動作する。

【００３４】図６に関連して下で説明するように、データ記憶装置４２０は、データ記憶装置４２０のセキュア部分に記憶することができるエンタイトルメントデータベース６００を備える。エンタイトルメントデータベース６００は顧客がエンタイトルメントを有するプログラムに対するプログラムキーk_pを得るために必要なキーツリー２００の部分を含む。また、データ記憶装置４２０はハッシュ関数H_0とH_1（４４０）を備える。また、図９に関連して下で説明するように、データ記憶装置４２０はデコードプロセス９００を含む。一般に、デコードプロセス９００は、プログラムキーk_pを得るために受信されたプログラム識別子ｐおよび記憶されたエンタイトルメント情報６００を用い、そしてプログラムを解読するためにプログラムキーk_pを用いて、顧客がエンタイトルメントを有するプログラムを解読する。

【００３５】図５は、ヘッドエンドサーバー３００によって送信される各プログラムｐ上に情報を記憶するプログラムデータベース５００を示している。この情報は、そのプログラムが属するパッケージおよび対応するプログラム識別子ｐとともに、例えば、課金期間に送信される。プログラムデータベース５００はレコード５０５～５２０のような複数のデコードを保持する。これらはそれぞれ異なるプログラムに関連づけられている。フィールド５２５にてプログラム名によって識別される各プログラム識別子に対して、プログラムデータベース５００は、フィールド５３０にてそのプログラムが属する対応するパッケージの指示を含み、フィールド５３５にて対応するプログラム識別子ｐを含む。

【００３６】図６は顧客がエンタイトルメントを有するプログラムに対してプログラムキーk_pを得るのに必要であるキーツリー２００の部分を含むエンタイトルメントデータベース６００を示している。前述したように、Ｔ（ｕ）はノードｕをルートとするサブツリー、すなわち、ノードｕのサブツリーにおけるリーフノード２４０～２４７に対応するプログラム識別子ｐのセットを表す。例えば、もし顧客がリーフノード２４０～２４３に対応する４つのプログラムを受信することに関してエンタイトルメントを有するならば、エンタイトルメント情報は、ノード２２０に対応する中間キーからなることとなる。この方法において、適切なハッシュ関数H_0とH_1（４４０）は必要に応じて、ノード２２０のサブツリーにおける各ノード２３０、２３２、２４０～２４３に対してプログラムキーk_pを得るために用いることができる。

【００３７】図６で示したエンタイトルメントデータベース６００は、リーフノード２４０～２４３に対応する四つのプログラムを受信する正規ユーザーであり（エンタイトルメントがある）、また、リーフノード２４６～２４７に対応する二つのプログラムを受信する正規ユーザーである。従って、エンタイトルメントデータベース６００に記録されたエンタイトルメント情報は、ノード２２０とノード２３６に対応する中間キーからなる。ノード２２０、２３６それぞれに対し、エンタイトルメントデータベース６００に記録されたエンタイトルメント情報はそれぞれ中間キー値k_{i0}とk_{i11}を有し、対応する部分的プログラム識別子ｐの指示を有する。顧客が選択したプログラムのパッケージに基づいてエンタイトル

メント情報配信プロセス７００によってエンタイトルメントデータベース６００が生成される方法は、図７と関連して下で説明する。

【００３８】プログラムパッケージング

本発明のツリー方式を用いて、様々なサイズの多くのプログラムのセットに対し小さなエンタイトルメントを確立することができる。パッケージされるプログラムの集合を用いてターゲットセットＳが確立される。サブツリーがターゲットセットＳを正確にカバーするようなツリーノードの最小セットを以下のように得る。

【数４】

$$T(S)=Z\subseteq T \quad \text{ただし、} \bigcup_{u\in Z} T(u)=S \text{、かつ、} |Z| \text{は最小であるように}$$

【００３９】パッケージＳに対するエンタイトルメント情報は、Ｔ（Ｓ）のノードにおいて保持される中間キーのセットk_iである。上で示すように、このキーのセットにより、セットトップターミナル４００が正確にＳ（のみ）におけるプログラムを解読する。原理的には、本発明のツリー方式は、いずれの任意のターゲットセットＳに対するエンタイトルメント情報をつくることができる。更に、しかし、もしプログラム識別子ｐが任意に割り当てられれば、エンタイトルメント情報はセットトップターミナル４００の制限されたセキュアメモリーにとって許されないほど大きくなってしまう。

【００４０】プロセス

上述のように、ヘッドエンドサーバー３００は図７に示したエンタイトルメント情報配信プロセス７００を実行し、正規ユーザーであるプログラムにアクセスするために各ユーザーにとって必要なエンタイトルメントデータベース６００を生成し配信する。前述のように、エンタイトルメントデータベース６００は顧客が正規ユーザーであるプログラムに対して、プログラムキーk_pを得るのに必要なキーツリー２００の各ノードに対して、対応する部分的プログラム識別子の指示および中間キー値k_iからなる。

【００４１】従って、エンタイトルメント情報配信プロセス７００はまず、顧客が選択したプログラムを識別する（７１０）。その後に、エンタイトルメント情報配信プロセス７００はツリーノードの最小セットＴ（Ｓ）を見つける。そのサブツリーは正確にターゲットセットＳをカバーする。ターゲットセットＳは、コンセキュティブプログラム識別子ｐの最大ディスジョイントインターバルへと分解される（７２０）。二つのプログラム識別子ｐは、そのバイナリー表現に対する整数がコンセキュティブである場合に、コンセキュティブと考えられる。

【００４２】そして、カバーＴ（Ｓ）が各インターバルに対して見つけられる（７３０）。中間キーのセットk_iと各インターバルに対するカバーＴ（Ｓ）のノードにて保持される対応する部分的プログラム識別子ｐが生成される（７４０）。最後に、生成されたエンタイトルメント情報がヘッドエンドサーバー３００によってセットトップターミナル４００へとダウンロードされ（７５０）、プログラム制御が終了する（７６０）。

【００４３】ターゲットセットＳにおけるインターバルの数はＩ（Ｓ）とすることができる。ｎのツリーノードのオーダーでプログラム識別子ｐの単一インターバルに対するカバーＴ（Ｓ）を計算するために、深さｎのキーツリー２００に問わなければならない。従って、エンタイトルメント情報配信プロセス７００の時間複雑さはＩ（Ｓ）・ｎのオーダーとなる。同様に、最小カバーＴ（Ｓ）の大きさは、Ｉ（Ｓ）・ｎのオーダーとなる。関連する内容のプログラムは効率的にそれらをパッケージングすることを可能にするプログラム識別子ｐが割り当てられるべきである。一例において、基本的なパッケージは、ビットプレフィックスμを有する全てのプログラム識別子ｐの形態である。

【００４４】このような単一トピックパッケージのエンタイトルメントは、キーツリー２００における単一のキーである。また、マルチトピックパッケージを副作用無しでアセンブルすることができる。エンタイトルメント情報は単に、マルチトピックスパッケージからなる個々のトピックスに対するキーのセットである。本発明に従い、プレフィックスμにより規定されるパッケージは同じ長さの０プレフィックスを用いてプログラムを解読するようにセットトップターミナル４００に対して強要しない。

【００４５】上述のように、ヘッドエンドサーバー３００は、図８に示すプログラム配信プロセス８００を実行し、プログラム識別子ｐを用いてプログラムを解読し送信するために、プログラムとマスターキーｍに割り当てられたプログラム識別子ｐに基づいてプログラムキーk_pを得る。プログラム配信プロセス８００は、実際の送信ステップ以外では、オフラインないし実時間で実行することは重要である。図８に示すように、プログラム配信プロセス８００は送信すべきプログラムを識別することによって本発明の原理を用いるプロセスを開始する（８１０）。

【００４６】その後に、プログラム配信プロセス８００はプログラムデータベース５００からのプログラムに対応するプログラム識別子ｐを取り出し（８２０）、そのプログラムに対応するプログラムキーk_pを計算する（８３０）。そしてプログラムは前のステップで計算されたプログラムキーk_pを用いて暗号化される（８４０）。最後に、プログラム配信プロセス８００は、プログラム識別子ｐとともに暗号化されたプログラムを送信し（８５０）、プログラム制御が終了する（８６０）。

【００４７】プログラム識別子ｐは、プログラム情報の

送信を通して、周期的にインターリーブされて送信することができ、プログラム時にチャネルを顧客が変更し、プログラムを解読するのに必要なプログラムキーk_pを得ることが可能とすることは重要である。別の実施例において、プログラム識別子ｐはBarkerチャネルのような別の制御チャネル上に連続的に送信することができる。

【００４８】上述のように、セットトップターミナル４００は図９に示したデコードプロセス９００を実行し、プログラムキーk_pを得るために記憶されたエンタイトルメント情報６００および受信されたプログラム識別子ｐを用いて、そのプログラムを解読するためにプログラムキーk_pを用いて顧客がエンタイトルメントされているプログラムを解読する。図９に示すように、デコードプロセス９００は特定のチャンネルにチューニングさせる顧客指示の受け取りの際に、本発明の原理を用いたプロセスを開始する（９１０）。

【００４９】その後に、セットトップターミナル４００は暗号化されたプログラムおよび送信されたプログラム識別子ｐを含む適切な信号を受信する（９２０）。デコードプロセス９００はエンタイトルメントデータベース６００から記憶されたエンタイトルメント情報を取り出す（９３０）。送信されたプログラムを含むかどうかを判断する（９４０）。もしステップ９４０にて受信プログラム識別子ｐの最左桁ビットに合致する部分プログラム識別子ｐを有するエントリーがエンタイトルメントデータベース６００にて存在しないと判断された場合、顧客には選択されたプログラムに対するエンタイトルメントはなく、プログラム制御は終了する（９８０）。

【００５０】しかし、もし受信されたプログラム識別子ｐの最左桁ビットに合致する部分プログラム識別子ｐを有するエンタイトルメントデータベース６００にエントリーが存在すれば、顧客には選択されたプログラムへのエンタイトルメントがある。従って、エンタイトルメントデータベース６００のエントリーから取り出した中間キーk_iを用いてプログラムキーk_pが計算される（９６０）。具体的には、プログラムキーk_pは以下のようにプログラム識別子ｐの（ｎ－ｒ）低いオーダーのビットのそれぞれの値が指示するように適切なハッシュ関数H_0またはH_1を作動させることによって計算される。

【数５】

$$K_p = H_{p_n}(\ldots H_{p_{r+1}}(H_{p_r}(K_I))\ldots)$$

【００５１】最後に、そのプログラムは得られたプログラムキーk_pを用いて解読され（９７０）、プログラム制御を終了する（９８０）。ここで、もし受信されたプログラムが顧客のエンタイトルメントの一部ではないような場合、送信プログラムとともに受信したプログラム識別子ｐの低位ビットに合致する部分的識別子ｐを有するエンタイトルメント情報がエンタイトルメントデータベース６００にはないことが重要である。

【００５２】また、デコードプロセス９００は解読キーを得たり、上述のように、顧客が要求チャンネルに対してエンタイトルメントがあるかどうかを判断する前に、顧客が特定のチャンネルを要求するのを待つことができ、また、デコードプロセス９００は代わりに、送信プログラム識別子ｐを得るために全てのチャネルを周期的にスキャンして、データ記憶装置４２０における記憶装置に対する解読キーを得て顧客のエンタイトルメントを予め判断することができることはまた重要である。

【００５３】適当なハッシュ関数

前述のように、もしハッシュ関数Ｈが擬似ランダムビット生成器であれば、ｐ→k_pのマッピングは擬似ランダム関数であることを証明できる。従って、もし実際のハッシュ関数Ｈが暗号学的に強ければ、暗号キーは予測することはできない。従って、もし海賊行為者が暗号化プログラムブロードキャストに対してのみアクセスを有するのであれば、本発明のツリー方式を用いて生成されたキーに関する知識では暗号を突破することはできないであろう。従って、ただ一つの関心事はビデオ暗号化アルゴリズムが公知のプレーンテキストアタックに対して対抗し得ることを確実にすることのみとなる。

【００５４】ハッシュ関数Ｈは二つの特性を保持すべきである。第１に、ハッシュ関数Ｈに対してイメージの半分$H_0(x)$または$H_1(x)$が与えられたとして入力ｘを計算することは難しくなければならないことがある。このことは、それら半分の両方のイメージを知っていたとしてもインバートすることが困難であるいずれの暗号学的ハッシュＨに対しても実際に成立する。また、H_1（ｘ）が知られていたとしてもH_0（ｘ）を計算することが困難でなければならず、逆も同様である。基本的には、関数Ｈをインバートすることが困難であっても、一方の半分のキーを知っていた場合は残りの半分のキーを完成させるのがより容易になる。もしそうであれば、ノードｕに対してプログラムk_pを知っている海賊行為者は、シブリング（同胞：sibling）ノードｖへのキーを計算することができ、そして、ノードｖのサブツリーにおける全てのプログラムへのキーを計算することができることとなる。

【００５５】本発明に従うツリー方式の一つの利点として、海賊行為をされたエンタイトルメントのマージを非効率的にすることがある。シブリングプログラムの対p_1、p_2およびそれらの親ノード有を考えてみる。海賊行為者がＨ（$k_p(u)$）の二つの半分である両方のプログラムp_1、p_2に対応するプログラムキーk_pを知っているものと想定する。海賊行為者はそれでもＨをインバートし、$k_p(u)$を計算することができない。なぜなら、Ｈは暗号法的ハッシュ関数であるからである。従って、マージされた海賊行為をされたエンタイトルメントは、コンパクトな$k_p(u)$ではなく、$k_p(p_1)$と$k_p(p_2)$の両方を含んでいなければならない。従って、チープ（であるが

異なる）エンタイトルメントを用いる複数のセットトップターミナル４００へと分けることは海賊行為者にとってよい戦略ではない。なぜなら、組合わさるエンタイトルメントは非常に大きくなってしまうからである。

【００５６】上述のように、適切な擬似ランダムハッシュ関数は、例えば、文献、O. Goldreich et al., "How to Construct Random Functions," J. ACM, 33:792-807 (1986)に記載されている。

【図面の簡単な説明】

【図１】本発明の一実施例に従って、暗号化されたプログラミング内容を送信するシステムを表すブロック図。

【図２】本発明に従うキーツリーの例を表す図。

【図３】図１のヘッドエンドサーバーのブロック図。

【図４】図１のセットトップターミナルのブロック図。

【図５】図３のプログラムデータベースからのテーブル。

【図６】図４のエンタイトルデータベースからのテーブル。

【図７】図３のヘッドエンドサーバが用いるエンタイトルメント情報配信プロセスを表す流れ図。

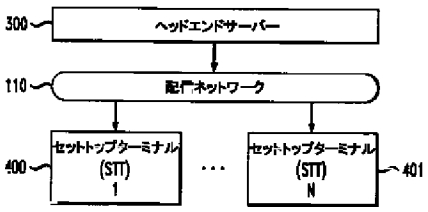
【図８】図３のヘッドエンドサーバが用いるプログラム配信流れ図を示すブロック図。

【図９】図４のセットトップターミナルが用いるレコードプロセスを表す流れ図。

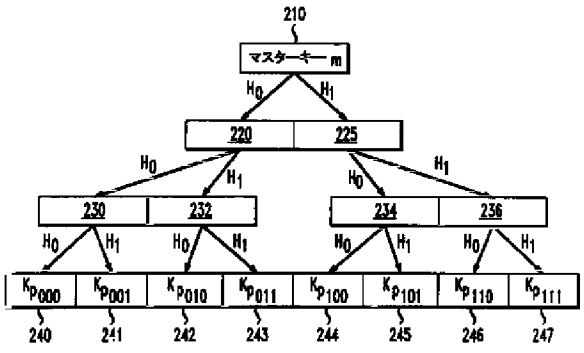
【符号の説明】

１１０　配信ネットワーク
２００　キーツリー
２２０、２３０、２３２、２３６、２４０～２４３、２４６～２４７　ノード
３００　ヘッドエンドサーバー
３１０、４１０　プロセッサー
３２０、４２０　データ記憶装置
３５０データベース
３３０、４３０　通信ポート
４００～４０１　セットトップターミナル
４４０　ハッシュ関数H_0とH_1
５００　プログラムデータベース
５０５～５２０　デコード
５２５、５３０、５３５　フィールド
６００　エンタイトルメントデータベース
７００　エンタイトルメント情報配信プロセス
７１０　顧客が選択したプログラムを識別
７２０　ターゲットセットコンセキュティブプログラム識別子ｐの最大ディスジョイントインターバルへと分解
７３０　各インターバルに対してカバーＴ（Ｓ）を見つける
７４０　各インターバルに対してカバーＴ（Ｓ）のノードにて、中間キーk_iのセットおよび対応する部分プログラム識別子ｐを生成
７５０　セットトップターミナルにエンタイトルメント情報を送信
７６０、８６０、９８０　終了
８００　プログラム配信プロセス
８１０　送信するべきプログラムを識別
８２０　プログラムデータベースからプログラム識別子ｐを取り出す
８３０　プログラムキーを計算
８４０　プログラムキーを用いてプログラムを暗号化
８５０　プログラム識別子ｐとともに暗号化されたプログラムを送信
９００　デコードプロセス
９１０　チャンネルにチューニングさせる顧客指示を取り出す
９２０　プログラムとプログラム識別子ｐを含む送信信号を受信
９３０　エンタイトルメントデータベースから記憶されたエンタイトルメント情報を取り出す
９４０　受信プログラム識別子ｐのＭＳＢに合致する部分プログラム識別子ｐを有するエントリーがあるか？
９６０　取り出したk_i値とハッシュ関数H_0とH_1を用いてプログラムキーk_pを計算
９７０　プログラムキーk_pを用いてプログラムを解読

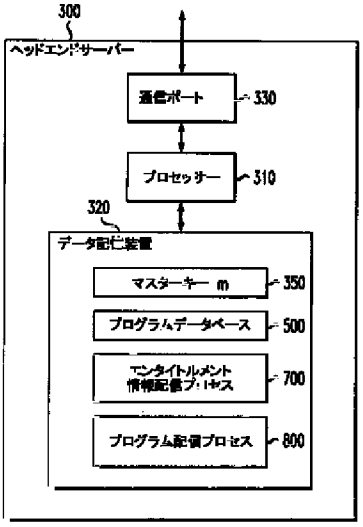
【図１】



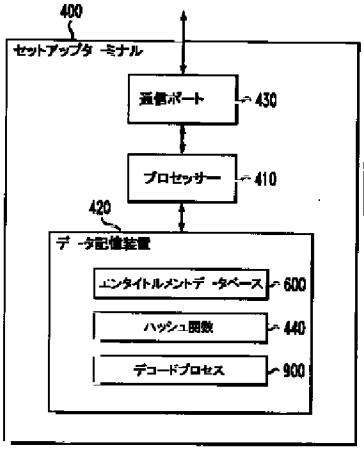
【図２】



【図3】



【図4】

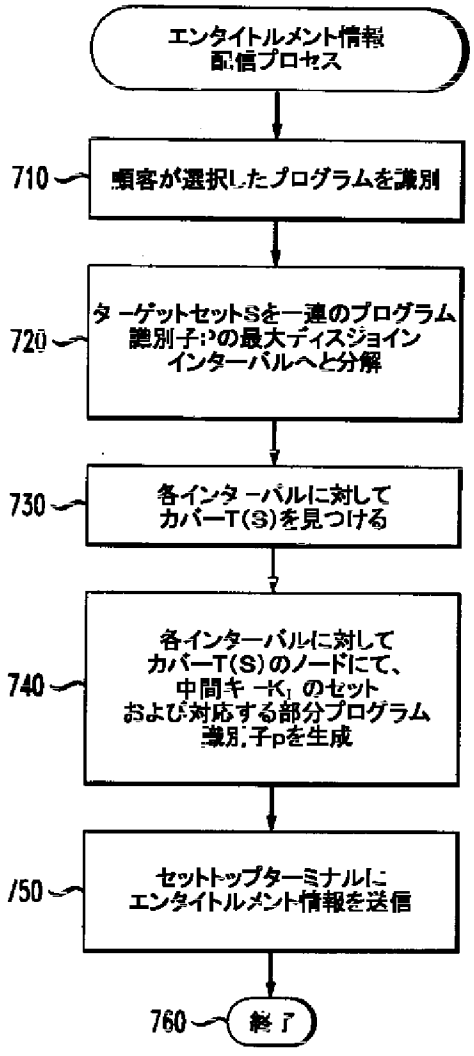


【図6】

エンタイトルメントデータベース 600

ノード	キー値	部分プログラム識別子
220	K_{I_0}	0
236	$K_{I_{11}}$	11

【図7】

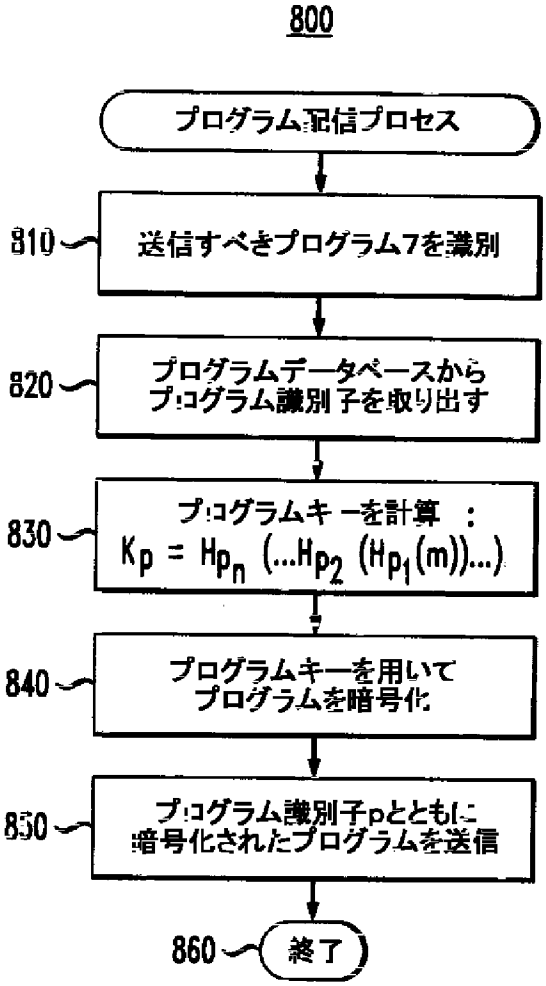


【図5】

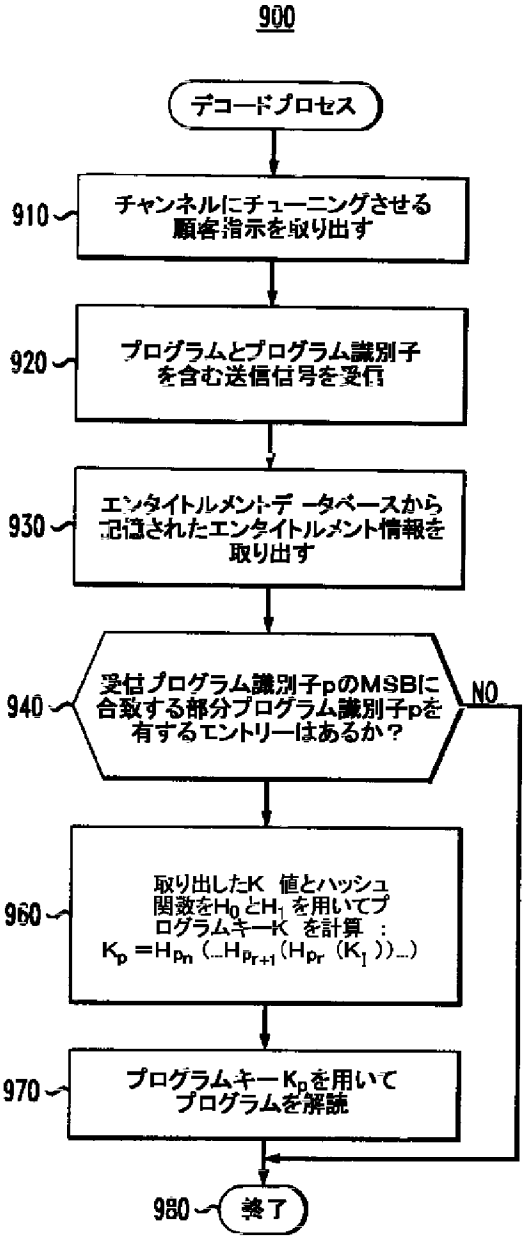
プログラムデータベース 500

	525 プログラム	530 パッケージ名	535 プログラム識別子
505	ワールドシリーズ試合	スポーツ、プロ野球、プレーオフ試合	p^1
510	スーパーボール	スポーツ、プロフットボール、プレーオフ試合	p^2
515	リ・ソンドオブミュージック	映画、ミュージカル	p^3
520	セサミストリート エピソード第1-54	子供向けプログラム、教育用プログラム	p^4

【図8】



【図9】



フロントページの続き

(51)Int.Cl.⁷	識別記号	FI		(参考)
H04N	7/16	H04N	7/08	Z
	7/167		7/167	Z

(71)出願人 596077259
600 Mountain Avenue, Murray Hill, New Jersey 07974-0636U. S. A.

(72)発明者 ダニエル ブライヘンバッハー
アメリカ合衆国、07901 ニュージャージー、スミット、スミット アベニュー 160

(72)発明者　アビシャイ　ウール
アメリカ合衆国、07901　ニュージャージー、リビングストン、フェルスウッド　45

【外国語明細書】

1. Title of Invention

Method and System For Transmitting A Program Having Restricted Access to An End-user

2. Claims

1. A method for transmitting a program having restricted access to an end-user, said method comprising the steps of:

assigning a program identifier to said program, said program identifier having a binary value;

defining at least one master key;

encrypting said program using a program key, said program key obtained by applying at least one hash function to said master key based on a binary value of said program identifier; and

transmitting said encrypted program together with said program identifier to said end-user.

2. The method according to claim 1, wherein said program identifier consists of *n* bits, and one of said hash functions is applied for each of the *n* bit positions of the program identifier depending on the corresponding bit value of the program identifier.

3. The method according to claim 1, further comprising the step of providing entitlement information to said end-users based on the set of programs obtained by said end-user

4. The method according to claim 3, wherein said entitlement information includes a portion of a key tree based on the set of programs obtained by said end-user.

5. The method according to claim 3, wherein said end-user uses said received program identifier to derive said program key from said stored entitlement information.

6. The method according to claim 1, wherein said program identifier is interleaved with the transmission of said encrypted program.

7. The method according to claim 1, wherein said program identifier is transmitted on a control channel.

8. A method for transmitting a program to a plurality of end-users, said method comprising the steps of:

encrypting said program using a program key, said program having a program identifier, said program key obtained by recursively applying a hash function to a master key based on the binary value of each bit position of said program identifier; and

transmitting said encrypted program and said program identifier to said end-user.

9. The method according to claim 8, wherein said program identifier consists of n bits, and a hash function is applied for each of the n bit positions of the program identifier depending on the corresponding bit value of the program identifier.

10. The method according to claim 8, further comprising the step of providing entitlement information to said end-users based on the set of programs obtained by said end-user

11. The method according to claim 10, wherein said entitlement information includes a portion of a key tree based on the set of programs obtained by said end-user.

12. The method according to claim 10, wherein said end-user uses said received program identifier to derive said program key from said stored entitlement information.

13. The method according to claim 8, wherein said program identifier is interleaved with the transmission of said encrypted program.

14. The method according to claim 8, wherein said program identifier is transmitted on a control channel.

15. A method for transmitting a program associated with at least one package of programs to a plurality of end-users, said method comprising the steps of:

providing entitlement information to said end-users based on the set of programs obtained by said end-user;

encrypting said program using a program key, said program having a program identifier, said program key obtained by recursively applying a hash function to a master key based on the binary value of each bit position of said program identifier; and

transmitting said program identifier with said encrypted program to said end-users, said end-users deriving said program key from said stored entitlement information if said end-user is entitled to said program.

16. The method according to claim 15, wherein said program identifier consists of *n* bits, and one of said hash functions is applied for each of the *n* bit positions of the program identifier depending on the corresponding bit value of the program identifier.

17. The method according to claim 15, wherein said entitlement information includes a portion of a key tree based on the set of programs obtained by said end-user.

18. The method according to claim 15, wherein said end-user uses said received program identifier to derive said program key from said stored entitlement information.

19. The method according to claim 15, wherein said program identifier is interleaved with the transmission of said encrypted program.

20. The method according to claim 15, wherein said program identifier is transmitted on a control channel.

21. A method for decoding an encrypted program, said method comprising the steps of:

receiving entitlement information from a provider of said program, said entitlement information including a portion of a key tree based on a set of programs obtained by said customer;

receiving said encrypted program and a program identifier, said encrypted program encrypted with a program key;

deriving said program key from said program identifier and said stored portion of said key tree; and

decrypting said encrypted program using said program key.

22. The method according to claim 21, wherein said program identifier consists of *n* bits, said master key is placed at the root of said key tree and said key tree is generated by applying a hash function to each node, until *n* tree levels have been created.

23. A method for decoding an encrypted program, said method comprising the steps of:

receiving entitlement information from a provider of said program, said entitlement information including at least one intermediate key from a key tree based on a set of programs obtained by said customer;

receiving said encrypted program and a program identifier, said encrypted program encrypted with a program key;

deriving said program key from said program identifier and said stored intermediate key by recursively applying a hash function to said intermediate key based on the binary value of said program identifier; and

decrypting said encrypted program using said program key.

24. The method according to claim 23, wherein said program identifier consists of n bits and said intermediate key corresponds to an intermediate node at a level r of said key tree, and wherein said hash function is applied to said intermediate key $n - r$ times.

25. A system for transmitting a program having restricted access to an end-user, said system comprising:

a memory for storing a master key and computer readable code; and

a processor operatively coupled to said memory, said processor configured to:

assign a program identifier to said program, said program identifier having a binary value;

define at least one master key;

encrypt said program using a program key, said program key obtained by applying at least one hash function to said master key based on a binary value of said program identifier; and

transmit said encrypted program together with said program identifier to said end-user.

26. A system for transmitting a program having restricted access to an end-user, said system comprising:

a memory for storing a master key and computer readable code; and

a processor operatively coupled to said memory, said processor configured to:

encrypt said program using a program key, said program having a program identifier, said program key obtained by recursively applying a hash function to

a master key based on the binary value of each bit position of said program identifier; and

transmit said encrypted program and said program identifier to said end-user.

27. A system for decoding an encrypted program, said system comprising:

a memory for storing a master key and computer readable code; and

a processor operatively coupled to said memory, said processor configured to:

receive entitlement information from a provider of said program, said entitlement information including a portion of a key tree based on a set of programs obtained by said customer;

receive said encrypted program and a program identifier, said encrypted program encrypted with a program key;

derive said program key from said program identifier and said stored portion of said key tree; and

decrypt said encrypted program using said program key.

28. An article of manufacture comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to assign a program identifier to a program, said program identifier having a binary value;

a step to define at least one master key;

a step to encrypt said program using a program key, said program key obtained by applying at least one hash function to said master key based on a binary value of said program identifier; and

a step to transmit said encrypted program together with said program identifier to said end-user.

29. An article of manufacture comprising:

a computer readable medium having computer readable code means embodied thereon, said computer readable program code means comprising:

a step to receive entitlement information from a provider of a program, said entitlement information including a portion of a key tree based on a set of programs obtained by said customer;

a step to receive said encrypted program and a program identifier, said encrypted program encrypted with a program key;

a step to derive said program key from said program identifier and said stored portion of said key tree; and

a step to decrypt said encrypted program using said program key.

3. Detailed Description of Invention

Field of the Invention

The present invention relates generally to a system for restricting access to transmitted programming content, and more particularly, to a system for transmitting an encrypted program together with a program identifier which is used by a set-top terminal, together with stored entitlement information, to derive the decryption key necessary to decrypt the program.

Background of the Invention

As the number of channels available to television viewers has increased, along with the diversity of the programming content available on such channels, it has become increasingly challenging for service providers, such as cable television operators and digital satellite service operators, to offer packages of channels and programs that satisfy the majority of the television viewing population. The development of packages that may be offered to customers is generally a marketing function. Generally, a service provider desires to offer packages of various sizes, from a single program to all the programs, and various combinations in between.

The service provider typically broadcasts the television programs from a transmitter, often referred to as the "head-end," to a large population of customers. Each customer is typically entitled only to a subset of the received programming, associated with purchased packages. In a wireless broadcast environment, for example, the transmitted programming can be received by anyone with an appropriate receiver, such as an antenna or a satellite dish. Thus, in order to restrict access to a transmitted program to authorized customers who have purchased the required package, the service provider typically encrypts the transmitted programs and provides the customer with a set-top terminal (STT) containing one or more decryption keys which may be utilized to decrypt programs that a customer is entitled to. In this manner, the set-top terminal

receives encrypted transmissions and decrypts the programs that the customer is entitled to, but nothing else.

In order to minimize piracy of the highly sensitive information stored in the set-top terminals, including the stored decryption keys, the set-top terminals typically contain a secure processor and secure memory, typically having a capacity on the order of a few kilobits, to store the decryption keys. The secure memory is generally non-volatile, and tamper-resistant. In addition, the secure memory is preferably writable, so that the keys may be reprogrammed as desired, for example, for each billing period. The limited secure memory capacity of conventional set-top terminals limits the number of keys that may be stored and thereby limits the number of packages which may be offered by a service provider. It is noted that the number of programs typically broadcast by a service provider during a monthly billing period can be on the order of 200,000.

In one variation, conventional set-top terminals contain a bit vector having a bit entry corresponding to each package of programs offered by the service provider. If a particular customer is entitled to a package, the corresponding bit entry in the bit vector stored in the set-top terminal is set to one ("1"). Thereafter, all programs transmitted by the service provider are encrypted with a single key. Upon receipt of a given program, the set-top terminal accesses the bit vector to determine if the corresponding bit entry has been set. If the bit entry has been set, the set-top terminal utilizes a single stored decryption key to decrypt the program. While, in theory, flexibility is achieved in the bit vector scheme by providing a bit entry for each package (a package generally consists of one program), the length of the bit vector would be impractical in a system transmitting many programs in a single billing period. In addition, access control in such a system is provided exclusively by the entries in the bit vector and is not cryptographic. Thus, if a customer is able to overwrite the bit vector, and set all bits to one ("1"), then the customer obtains access to all programs.

In a further variation, programs are divided into packages, and all programs in a given package are encrypted using the same key. Again, each package typically corresponds to one television channel. The set-top terminal stores a decryption key for

each package the customer is entitled to. Thus, if a program is to be included in a plurality of packages, then the program must be retransmitted for each associated package, with each transmission encrypted with the encryption key corresponding to the particular package. Although the access control is cryptographic, the overhead associated with retransmitting a given program a number of times discourages service providers from placing the same program in a number of packages and thereby limits flexibility in designing packages of programs.

While such previous systems for encrypting and transmitting programming content have been relatively successful in restricting access to authorized customers, they do not permit a service provider, such as a television network, to offer many different packages containing various numbers of programs to customers, without exceeding the limited secure memory capacity of the set-top terminal or significantly increasing the overhead. United States Patent Application Serial Number 08/912,186, filed August 15, 1997 and assigned to the assignee of the present invention, hereinafter referred to as the "Vspace System," discloses a cryptographic method and apparatus for restricting access to transmitted programming content.

Each program in the Vspace System is encrypted by the head-end server prior to transmission, using a program key, K_P. Each of the program keys is a linear combination of a defined set of master keys, M. A program identifier identifying the program is transmitted with the encrypted programming content. The customer's set-top terminal can derive the decryption key from only the received program identifier, p, and previously stored entitlement information. The Vspace System provides a cryptographic access control mechanism, while permitting flexible packages (since the program does not need to be retransmitted for each associated package) without significantly extending the program header (only the program identifier is transmitted with the program).

Summary of the Invention

Generally, encrypted programming content is transmitted by a service provider using a transmitter, or head-end server, to one or more customers. According to one

aspect of the invention, a program identifier, p, used to identify the program is transmitted to the customer with the programming content. Each customer has a set-top terminal or another mechanism to restrict access to the transmitted multimedia information using decryption keys. The set-top terminal receives entitlement information from the head-end, corresponding to one or more packages of programs that the customer is entitled to for a given period.

Each program is encrypted by the head-end server prior to transmission, using a program key, K_P, which may be unique to the program. In addition to transmitting the encrypted program, the head-end server transmits the program identifier, p, to the set-top terminal. The set-top terminal uses the received program identifier, p, together with the stored entitlement information, to derive the decryption key necessary to decrypt the program. In this manner, if a customer is entitled to a particular program, the set-top terminal will be able to derive the encrypted program key, K_P, using the stored and received information, and thereafter use the program key, K_P, to decrypt the encrypted program. In various embodiments, the program identifier, p, can be interleaved with the program portion or transmitted on a separate dedicated control channel.

According to one aspect of the invention, each of the k-bit program keys, K_P, used to encrypt transmitted programs is obtained by applying one or more pseudo-random hash functions, H, to a master key, m. In one implementation, a length-doubling hash function, H, is utilized. Thus, the hash function, H, takes a k-bit binary value and produces a binary value having a length of $2k$. The output of the hash function, H, can be represented as a pair of k-bit binary values, H_0 and H_1, where H_0 is referred to as the left half of the output of the hash function, and H_1 is the right half of the output of the hash function.

In an illustrative implementation, a program key, K_p, is obtained by recursively applying a hash function, H_0 or H_1, to the master key, m, depending on the corresponding binary value of each bit position of the program identifier, p. Thus, if the program identifier, p, consists of n bits, one of the hash functions, H_0 or H_1, is applied for each of the n bit positions of the program identifier, p, depending on the

corresponding bit value of the program identifier, p. Initially, one of the hash functions, H_0 or H_1, is applied to the master key, m, depending on the binary value of the most significant bit of the program identifier, p. Thereafter, for each of the remaining (n-1) bit positions, one of the hash functions, H_0 or H_1, is applied to the result of the previous hash operation, depending on the binary value of the corresponding bit. The calculation of the program key, K_p can be represented as follows:

$$K_p = H_{p_n}(...H_{p_2}(H_{p_1}(m))...).$$

The hash operation can be represented in terms of an *n*-level binary tree, T, referred to as the key tree, with the master key, *m*, placed at the root of the tree. The tree is generated by applying the hash functions H_0 and H_1 to each node, until the desired number of tree levels (n) have been created. The program keys, K_p, correspond to the leaf nodes at the bottom level of the tree. The binary index (and likewise the program identifiers, p) associated with each program key, K_p, corresponds to the path through the key tree from the root to the desired leaf node. Thus, the index or label of a given node, *u*, is the concatenation of the labels on the edges on the path from the root to the node *u*. *T(u)* denotes the subtree rooted at node *u*, or the set of program identifiers, p, corresponding to the leaves in the subtree of node *u*. For an internal node, *u*, at depth r in the key tree, with a partial program identifier, p, $(u_1,..., u_r)$, the keys of any program in the subtree T(u) can be computed by activating the hash function $n - r$ times.

A more complete understanding of the present invention, as well as further features and advantages of the present invention, will be obtained by reference to the following detailed description and drawings.

Brief Description of the Drawings

FIG. 1 is a schematic block diagram illustrating a system for transmitting encrypted programming content in accordance with one embodiment of the present invention;

FIG. 2 is a conceptual representation of an exemplary key tree in accordance with the present invention;

FIG. 3 is a schematic block diagram of an exemplary head-end server of FIG. 1;

FIG. 4 is a schematic block diagram of an exemplary set-top terminal of FIG. 1;

FIG. 5 illustrates a sample table from the program database of FIG. 3;

FIG. 6 illustrates a sample table from the entitlement database of FIG. 4;

FIG. 7 is a flow chart describing an exemplary entitlement information distribution process as implemented by the head-end server of FIG. 3;

FIG. 8 is a flowchart describing an exemplary program distribution process as implemented by the head end server of FIG. 3; and

FIG. 9 is a flowchart describing an exemplary decode process as implemented by the set-top terminal of FIG. 4.

Detailed Description

FIG. 1 shows an illustrative network environment for transferring encrypted multimedia information, such as video, audio and data, from a service provider using a transmitter, such as a head-end server 300, discussed further below in conjunction with FIG. 3, to one or more customers having set-top terminals 400-401, such as the set-top terminal 400, discussed further below in conjunction with FIG. 4, over one or more distribution networks 110. As used herein, a set-top terminal includes any mechanism to restrict access to the transmitted multimedia information using decryption keys, including, for example, a computer configuration or a telecommunications device. It is possible for software executed by the set-top terminal to be downloaded by the service provider. The distribution network 110 can be a wireless broadcast network for distribution of programming content, such as a digital satellite service ("DSS™"), or a conventional wired network, such as the cable television network ("CATV"), the Public

Switched Telephone Network ("PSTN"), an optical network, a broadband integrated services digital network ("ISDN") or the Internet.

According to a feature of the present invention, the set-top terminal 400 intermittently receives entitlement information from the head-end server 300, which permits a customer to access programs that the customer is entitled to for a given time interval, such as a billing period. As used herein, a package is a predefined set of programs, and a given program can belong to one or more packages. A program is any continuous multimedia transmission of a particular length, such as a television episode or a movie. The entitlement information can be downloaded from the head-end server 300 to the set-top terminal 400 using any suitably secure uni-directional or bi-directional protocol, as would be apparent to a person of ordinary skill.

PROGRAM KEYS AND PROGRAM IDENTIFIERS

As discussed further below, each transmitted program is encrypted by the head-end server 300 using a program key, K_P, which may be unique to the program. For a detailed discussion of suitable encryption and security techniques, see B. Schneier, Applied Cryptography (2d ed. 1997), incorporated by reference herein. In addition to transmitting the encrypted program, the head-end server 300 also transmits an *n*-bit program identifier, p, to the set-top terminals 400, which may be utilized by the set-top terminal 400, together with stored entitlement information, to derive the decryption key necessary to decrypt the program, in a manner described further below. As discussed below in a section entitled ASSIGNING PROGRAM IDENTIFIERS TO PROGRAMS, the program identifiers, p, are not chosen arbitrarily. In one preferred embodiment, the program identifier, p, consists of a thirty-two (32) bit value that may be transmitted, for example, in the ECM field defined in the MPEG-2 standard. In this manner, if a customer is entitled to a particular program, the set-top terminal 400 will be able to derive the program key, K_P, from stored and received information, and thereafter use the program key, K_P, to decrypt the encrypted program.

According to a further feature of the present invention, each of the *k*-bit program keys, K_P, used to encrypt transmitted programs is obtained by applying one or more pseudo-random hash functions to a master key, *m*. For a detailed discussion of suitable pseudo-random hash functions, see, for example, O. Goldreich et al., "How to Construct Random Functions," J. ACM, 33:792-807 (1986), incorporated by reference herein.

In one implementation, a crytographically-secure, length doubling, hash function is utilized, as follows:

$$H : \{0,1\}^k \rightarrow \{0,1\}^{2k},$$

where, *k* is the length of the program key, K_p. Thus, the hash function, H, takes a *k*-bit binary value and produces a binary value having a length of 2*k*. The output of the hash function, H, can be represented as a pair of *k*-bit binary values, H_0 and H_1, where H_0 is referred to as the left half of the output of the hash function, H (most significant bits), and H_1 is the right half of the output of the hash function, H (most significant bits). H_0 and H_1 can be said to be separate hash functions. In one illustrative implementation, when *k* equals 160, H could be defined by using the secret hash standard, SHA-1, as defined in Secure Hash Standard, National Institute of Standards and Technology, NIST FIPS PUB 180-1, U.S. Dept. of Commerce (April, 1995), incorporated by reference herein. In other words, H_0 equals SHA-1 ($x\|0$), and H_1 equals SHA-1 ($x\|1$), where 0 and 1 are all-zero and all-one bit strings, respectively.

According to a further feature of the present invention, a program key, K_p, is obtained by recursively applying one or more hash functions to the master key, m, depending on the binary value of the program identifier, p. In one implementation, the program key, K_p, is obtained by recursively applying one of the hash functions, H_0 or H_1, to the master key, m, depending on the binary value of each bit position of the program identifier, p. Generally, if the program identifier, p, consists of *n* bits, one of the hash functions, H_0 or H_1, is applied for each of the *n* bit positions of the program identifier, p, (starting with the most significant bit) depending on the corresponding bit value of the program identifier, p. Initially, one of the hash functions, H_0 or H_1, is applied to the

master key, m, depending on the binary value of the most significant bit. Thereafter, for each of the remaining (n-1) bit positions, one of the hash functions, H_0 or H_1, is applied to the result of the previous hash operation, depending on the binary value of the corresponding bit. As discussed below in a section entitled THE KEY TREE, the hash operation can be represented as follows:

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots).$$

As previously indicated, the head-end server 300 will transmit the program identifier, p, with the encrypted program. Thus, given the program identifier, p, the set-top terminal 400 must obtain the program key, K_P, used to decrypt the received program. As previously indicated, the program key, K_P, is obtained by recursively applying one or more hash functions to a master key, m, depending on the binary value of the program identifier, p. The program keys, K_P, must be obtained by the customer's set-top terminal 400 indirectly using the stored entitlement information, discussed below, and the received program identifier, p.

THE KEY TREE

As previously indicated, a program key, K_p, is obtained by recursively applying one or more hash functions, H, to a master key, m, depending on the binary value of the program identifier, p. A single *k*-bit master key, *m*, is utilized. The bits of the program identifier, p, are denoted by $p = (p_1,\ldots,p_n)$, where p_1 is the most significant bit and p_n is the least significant bit. According to a feature of the present invention, the encryption key, K_p, for a program with a program identifier, p, is defined as follows:

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots).$$

The hash operation can also be represented in terms of a full *n*-level binary tree T, referred to as the key tree 200, shown in FIG. 2. The illustrative key tree 200, shown in FIG. 2, corresponds to an implementation having program identifiers, p, consisting of three bits. As shown in FIG. 2, the master key, *m*, is placed at the root 210 of the tree 200. The program keys, K_p, correspond to the leaf nodes, such as the leaf nodes 240-

247. The index associated with each program key, K_p, shown in FIG. 2, such as the index 011 associated with the program key, K_p, of the leaf node 243, indicates the path through the key tree 200 from the root 210 to the leaf node 243. For example, the program key, K_p, of the leaf node 243 is obtained by following a left edge (H_0) from the root 210, a right edge (H_1) from the node 220 and a right edge (H_1) from the node 232. In other words, H_0 is initially applied to the master key, m, then H_1 is applied to a first hash result, and H_1 is again applied to the second hash result. The resulting value is the program key, K_{p011}.

Thus, the label of a given node, *u*, such as the node 243, is the concatenation of the labels on the edges on the path from the root 210 to the node *u*. The label of each node can be identified with the program identifiers, p. *T(u)* is utilized to denote the subtree rooted at node *u*, or equivalently, to denote the set of program identifiers, p, corresponding to the leaves in the subtree of node *u*. For an internal node, *u*, at depth r in the key tree 200, with a partial program identifier, p, $(u_1,\ldots, u_r)$, the keys of any program in the subtree T(u) can be computed. The key of any program in the subtree of node *u* is computed by activating the hash function $n - r$ times. Specifically, the appropriate hash function, H_0 or H_1, is utilized as directed by the value of each of the $n - r$ low order bits of the program identifier, p. Thus, the program key, K_p, corresponding to a node *u* can serve as an entitlement for all programs in the subtree of node *u*.

If the function H is a pseudo-random generator, then the mapping of the program keys, K_p $\{0,1\}^n \rightarrow \{0,1\}^k$, parameterized by the master key, m, is a pseudo-random function. See, for example, O. Goldreich et al., "How to Construct Random Functions," J. ACM, 33:792-807 (1986), incorporated by reference above.

SYSTEM COMPONENTS

FIG. 3 is a block diagram showing the architecture of an illustrative head-end server 300. The head end may be associated with a television network, a cable operator, a digital satellite service operator, or any service provider transmitting encrypted programming content. The head-end server 300 may be embodied, for example, as an

RS 6000 server, manufactured by IBM Corp., as modified herein to execute the functions and operations of the present invention. The head-end server 300 includes a processor 310 and related memory, such as a data storage device 320. The processor 310 may be embodied as a single processor, or a number of processors operating in parallel. The data storage device 320 and/or a read only memory (ROM) are operable to store one or more instructions, which the processor 310 is operable to retrieve, interpret and execute.

As discussed above, the data storage device 320 includes a master key database 350 for storing the master key, *m*. The master key, m, may be updated, for example, once per billing period. In addition, as discussed further below in conjunction with FIG. 5, the data storage device 320 includes a program database 500. The program database 500 indicates the program identifier, p, and associated packages corresponding to each program. In addition, as discussed further below in conjunction with FIGS. 7 AND 8, the data storage device 320 includes an entitlement information distribution process 700 and a program distribution process 800. Generally, the entitlement information distribution process 700 generates and distributes the entitlement information required by each customer to access entitled programs. In addition, the program distribution process 800 derives the program key, K_p, based on the program identifier, p, assigned to the program in order to encrypt and transmit the program with the program identifier, p.

The communications port 330 connects the head-end server 300 to the distribution network 110, thereby linking the head-end server 300 to each connected receiver, such as the set-top terminal 400 shown in FIG. 1.

FIG. 4 is a block diagram showing the architecture of an illustrative set-top terminal 400. The set-top terminal 400 may be embodied, for example, as a set-top terminal (STT) associated with a television, such as those commercially available from General Instruments Corp., as modified herein to execute the functions and operations of the present invention. The set-top terminal 400 includes a processor 410 and related memory, such as a data storage device 420, as well as a communication port 430, which operate in a similar manner to the hardware described above in conjunction with FIG. 3.

As discussed further below in conjunction with FIG. 6, the data storage device 420 includes an entitlement database 600 that may be stored in a secure portion of the data storage device 420. The entitlement database 600 includes those portions of the key tree 200 that are necessary to derive the program keys, K_p, for the programs to which the customer is entitled. In addition, the data storage device 420 includes the hash functions, H_0 and H_1, 440. In addition, as discussed further below in conjunction with FIG. 9, the data storage device 420 includes a decode process 900. Generally, the decode process 900 decrypts programs that a customer is entitled to, by using the received program identifier, p, and the stored entitlement information 600 to derive the program key, K_P, and then using the program key, K_P, to decrypt the program.

FIG. 5 illustrates an exemplary program database 500 that stores information on each program, p, which will be transmitted by the head-end server 300, for example, during a given billing period, including the packages the program belongs to and the corresponding program identifier, p. The program database 500 maintains a plurality of records, such as records 505-520, each associated with a different program. For each program identified by program name in field 525, the program database 500 includes an indication of the corresponding packages to which the program belongs in field 530 and the corresponding program identifier, p, in field 535.

FIG. 6 illustrates an exemplary entitlement database 600 that includes those portions of the key tree 200 that are necessary to derive the program keys, K_p, for the programs to which the customer is entitled. As previously indicated, *T(u)* is utilized to denote the subtree rooted at a node *u*, or equivalently, to denote the set of program identifiers, p, corresponding to the leaf nodes 240-247 in the subtree of node *u*. For example, if a customer is entitled to receive the four programs corresponding to the leaf nodes 240-243, the entitlement information would consist of the intermediate key associated with node 220. In this manner, the appropriate hash functions, H_0 and H_1, 440 can be used to derive the program keys, K_p, for each node 230, 232, 240-243 in the subtree of node 220, as necessary.

The exemplary entitlement database 600 shown in FIG. 6 corresponds to a customer that is entitled to receive the four programs corresponding to the leaf nodes 240-243, as well as the two programs corresponding to the leaf nodes 246-247. Thus, the entitlement information recorded in the entitlement database 600 consists of the intermediate keys associated with node 220 and node 236. For each node 220 and 236, the entitlement information recorded in the entitlement database 600 includes the intermediate key value, K_{I_0} and $K_{I_{11}}$, respectively, and an indication of the corresponding partial program identifier, p. The manner in which the entitlement information 600 is generated by the entitlement information distribution process 700 based on packages of programs selected by a customer is discussed below in conjunction with FIG. 7.

PROGRAM PACKAGING

Small entitlements can be established for many sets of programs of varying size, using the tree scheme of the present invention. A target set, S, is established using the collection of programs to be packaged. A minimal set of tree nodes is obtained whose subtrees precisely cover the target set, S, as follows:

$$T(S) = Z \subseteq T \text{ such that } \bigcup_{u \in Z} T(u) = S, \text{ and } |Z| \text{ is minimal.}$$

The entitlement information for the package, S, is the set of intermediate keys, K_I, held at the nodes of T(S). As indicated above, this set of keys allows the set-top terminal 400 to decrypt exactly the programs in S but nothing else. It is noted that, in principle, the tree scheme of the present invention can create entitlement information for any arbitrary target set, S. It is further noted, however, that if the program identifiers, p, are assigned arbitrarily then the entitlement information may become prohibitively large for the limited secure memory of the set-top terminals 400.

PROCESSES

As discussed above, the head-end server 300 executes an entitlement information distribution process 700, shown in FIG. 7, to generate and distribute the entitlement

information 600 required by each customer to access entitled programs. As previously indicated, the entitlement information 600 consists of the intermediate key value, K_I, and an indication of the corresponding partial program identifier, p, for each node of the key tree 200 that is necessary to derive the program keys, K_p, for the programs to which the customer is entitled.

Thus, the entitlement information distribution process 700 initially identifies the programs selected by the customer during step 710. Thereafter, the entitlement information distribution process 700 finds a minimal set of tree nodes, T(S), whose subtrees precisely cover the target set, S. The target set, S, is decomposed during step 720 into maximal disjoint intervals of consecutive program identifiers, p. It is noted that two program identifiers, p, are considered consecutive if the integers corresponding to the binary representations are consecutive. A cover, T(S), is then found for each interval during step 730. The set of intermediate keys, K_I, and corresponding partial program identifiers, p, held at the nodes of the cover, T(S), for each interval are generated during step 740. Finally, the generated entitlement information is downloaded by the head-end server 300 to the set-top terminal 400 during step 750, before program control terminates during step 760.

The number of intervals in the target set, S, is referred to as I(S). To compute a cover, T(S), for a single interval of program identifiers, p, on the order of n tree nodes must be visited in a key tree 200 of depth n. Thus, the time complexity of the entitlement information distribution process 700 is on the order of $I(S) \cdot n$. Likewise, the size of the minimal cover, T(S), is on the order of $I(S) \cdot n$. Programs with related content should be assigned program identifiers, p, that allow them to be packaged efficiently. In one implementation, basic packages are of the form all program identifiers, p, with a bit prefix μ. An entitlement for such a single-topic package is a single key in the key tree 200. Moreover, multi-topic packages can be assembled with no side-effects. The entitlement information is simply the set of keys for the individual topics that comprise the multi-topic package. In accordance with the present invention, a package defined by

a prefix μ does not allow the set-top terminal 400 to decrypt programs with a 0 prefix of the same length.

As discussed above, the head-end server 300 executes a program distribution process 800, shown in FIG. 8, to derive the program key, K_P, based on the program identifier, p, assigned to the program and the master key, *m*, in order to encrypt and transmit the program with the program identifier, p. It is noted that the program distribution process 800, other than the actual transmission step, can be executed offline or in real-time. As illustrated in FIG. 8, the program distribution process 800 begins the processes embodying the principles of the present invention during step 810 by identifying a program to be transmitted.

Thereafter, the program distribution process 800 retrieves the program identifier, p, corresponding to the program from the program database 500, during step 820, and then calculates the program key, K_P, corresponding to the program during step 830. The program will then be encrypted during step 840 with the program key, K_P, calculated during the previous step. Finally, the program distribution process 800 will transmit the encrypted program together with the program identifier, p, during step 850, before program control terminates during step 860. It is noted that the program identifier, p, can be transmitted periodically interleaved throughout the transmission of the program information, so that a customer can change channels during a program and be able to derive the program key, K_P, which is required to decrypt the program. In an alternate embodiment, the program identifier, p, can be continuously transmitted on a separate control channel, such as a Barker channel.

As discussed above, the set-top terminal 400 executes a decode process 900, shown in FIG. 9, to decrypt programs that a customer is entitled to, by using the received program identifier, p, and the stored entitlement information 600 to derive the program key, K_P, and then using the program key, K_P, to decrypt the program. As illustrated in FIG. 9, the decode process 900 begins the processes embodying the principles of the present invention during step 910, upon receipt of a customer instruction to tune to a particular channel.

Thereafter, the set-top terminal 400 will receive the appropriate signal during step 920, including the encrypted program and the transmitted program identifier, p. The decode process 900 then retrieves the stored entitlement information from the entitlement database 600 during step 930. A test is performed during step 940 to determine if with the transmitted program. If it is determined during step 940 that an entry does not exist in the entitlement database 600 having a partial program identifier, p, that matches the most significant bits of the received program identifier, p, then the customer is not entitled to the selected program and program control terminates during step 980.

If, however, an entry does exist in the entitlement database 600 having a partial program identifier, p, that matches the most significant bits of the received program identifier, p, then the customer is entitled to the selected program. Thus, the program key, K_P, is then calculated during step 960 using the intermediate key, K_I, retrieved from the entry of the entitlement database 600. Specifically, the program key, K_p is computed by activating the appropriate hash function, H_0 or H_1, as directed by the value of each of the $n - r$ low order bits of the program identifier, p, as follows:

$$K_p = H_{p_n}(\ldots H_{p_{r+1}}(H_{p_r}(K_I))\ldots).$$

Finally, the program is decrypted using the derived program key, K_P, during step 970, before program control terminates during step 980. It is noted that if the received program is not part of the customer's entitlement, then there is no entitlement information in the entitlement database 600 having a partial program identifier, p, that matches the low order bits of the program identifier, p, received with the transmitted program.

It is further noted that the decode process 900 can wait for the customer to request a particular channel before attempting to derive the decryption keys and determine whether the customer is entitled to the requested channel, as described above, or the decode process 900 can alternatively periodically scan all channels to obtain the

transmitted program identifiers, p, in order to derive the decryption keys for storage in the data storage device 420 and predetermine the customer's entitlement.

SUITABLE HASH FUNCTIONS

As previously indicated, if the hash function, H, is a pseudo-random bit generator, then the mapping of $p \rightarrow K_p$ is provably a pseudo-random function. Thus, if the actual hash function, H, is cryptographically strong, then the encryption keys would be unpredictable. Accordingly, if a pirate only has access to the encrypted program broadcast, the knowledge that the keys were generated using the tree scheme of the present invention does not seem to help in breaking the encryption. Therefore, essentially the only concern is to ensure that the video encryption algorithm can withstand known plaintext attacks.

The hash function, H, should possess two properties. First, it must be hard to compute the input x given half of the image $H_0(x)$ or $H_1(x)$ for the hash function, H. This certainly holds for any cryptographic hash H, which is hard to invert even when both halves of the image are known. In addition, it must be hard to compute $H_0(x)$ even when $H_1(x)$ is known, and vice versa. In principle, it may be easier to complete a missing half-key when the other half is known, even if the function H is hard to invert. If this is the case, then a pirate who knows the program key, K_p for a node u may be able to compute the key to a sibling node, v, and then to all the programs in the subtree of node v.

One advantage of the tree scheme in accordance with the present invention is that it makes merging pirated entitlements inefficient. Consider a pair of sibling programs, p_1 and p_2, and their parent node, u. Suppose that the pirate knows the program key, K_p, corresponding to both programs, p_1 and p_2, which are the two halves of $H(K_p(u))$. The pirate still cannot invert H and compute $K_p(u)$ since H is a cryptographic hash function. Thus, the merged pirated entitlements would have to contain both $K_p(p_1)$ and $K_p(p_2)$, rather than more compact $K_p(u)$. Thus, breaking into multiple set-top terminals 400 with

cheap (but different) entitlements is not a good strategy for the pirate, since the combined entitlement will be very large.

As previously indicated, suitable pseudo-random hash functions are discussed, for example, in O. Goldreich et al., "How to Construct Random Functions," J. ACM, 33:792-807 (1986), incorporated by reference above.

It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.

4. Brief Description of Drawings

Written above.

FIG. 1

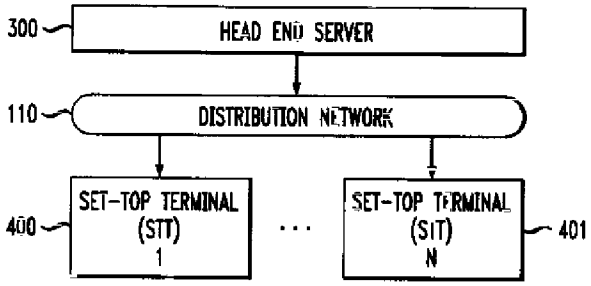

300
HEAD END SERVER
110
DISTRIBUTION NETWORK
400
SET-TOP TERMINAL (STT) 1
SET-TOP TERMINAL (SIT) N
401

FIG. 2

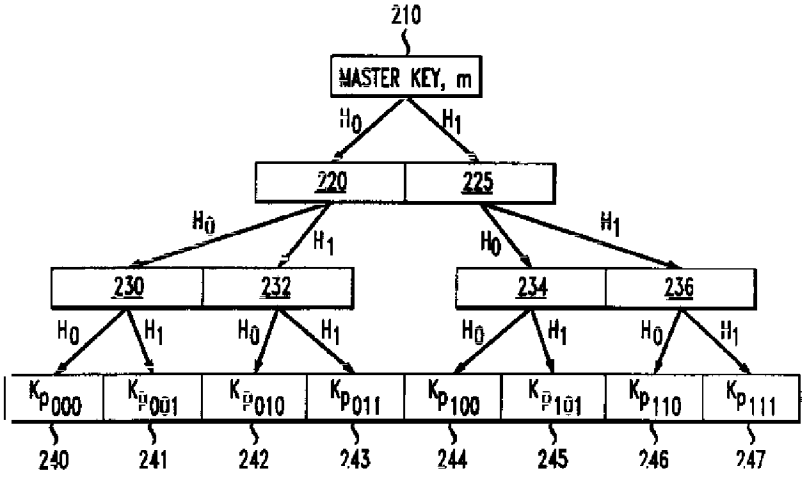

210
MASTER KEY, m
220
225
230
232
234
236
240
241
242
243
244
245
246
247

FIG. 3

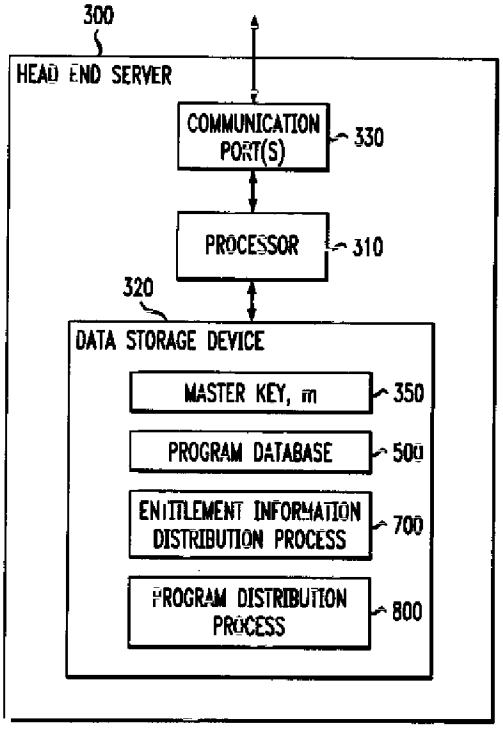

300
HEAD END SERVER
COMMUNICATION PORT(S)
330
PROCESSOR
310
320
DATA STORAGE DEVICE
MASTER KEY, m
350
PROGRAM DATABASE
500
ENTITLEMENT INFORMATION DISTRIBUTION PROCESS
700
PROGRAM DISTRIBUTION PROCESS
800

FIG. 4

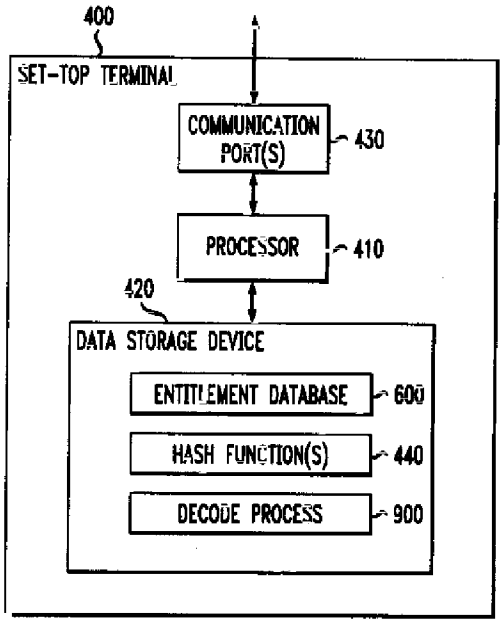

400
SET-TOP TERMINAL
COMMUNICATION PORT(S)
430
PROCESSOR
410
420
DATA STORAGE DEVICE
ENTITLEMENT DATABASE
600
HASH FUNCTION(S)
440
DECODE PROCESS
900

FIG. 5

PROGRAM DATABASE 500

	525 PROGRAM	530 PACKAGE NAMES	535 PROGRAM IDENTIFIER
505	WORLD SERIES GAME 5	SPORTS, PROFESSIONAL BASEBALL, PLAYOFF GAMES	P^1
510	SUPER BOWL	SPORTS, PROFESSIONAL FOOTBALL, PLAYOFF GAMES	P^2
515	SOUND OF MUSIC	MOVIES, MUSICALS	P^3
520	SESAME STREET, EPISODE NO. 554	CHILDREN'S PROGRAMMING; EDUCATIONAL PROGRAMMING	P^4

FIG. 6

ENTITLEMENT DATABASE 600

NODE	KEY VALUE	PARTIAL PROGRAM IDENTIFIER, p
220	K_{I_0}	0
236	$K_{I_{11}}$	11

FIG. 7

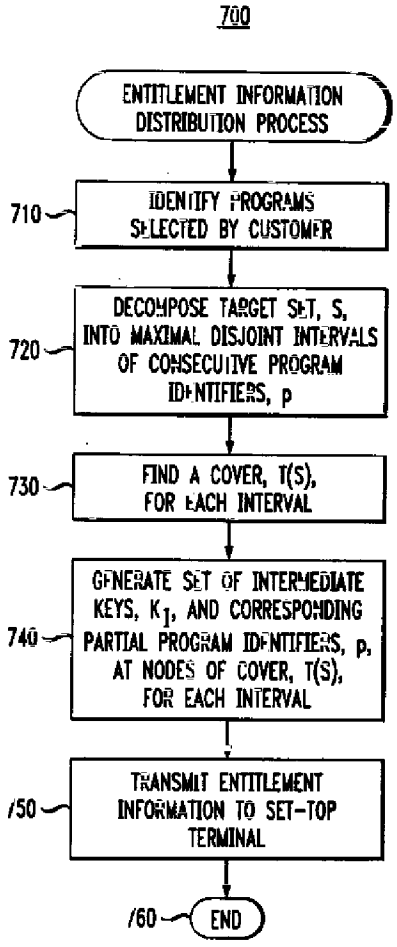

700
ENTITLEMENT INFORMATION DISTRIBUTION PROCESS
710
IDENTIFY PROGRAMS SELECTED BY CUSTOMER
720
DECOMPOSE TARGET SET, S, INTO MAXIMAL DISJOINT INTERVALS OF CONSECUTIVE PROGRAM IDENTIFIERS, p
730
FIND A COVER, T(S), FOR EACH INTERVAL
740
GENERATE SET OF INTERMEDIATE KEYS, K_1, AND CORRESPONDING PARTIAL PROGRAM IDENTIFIERS, p, AT NODES OF COVER, T(S), FOR EACH INTERVAL
750
TRANSMIT ENTITLEMENT INFORMATION TO SET-TOP TERMINAL
760
END

FIG. 8

800

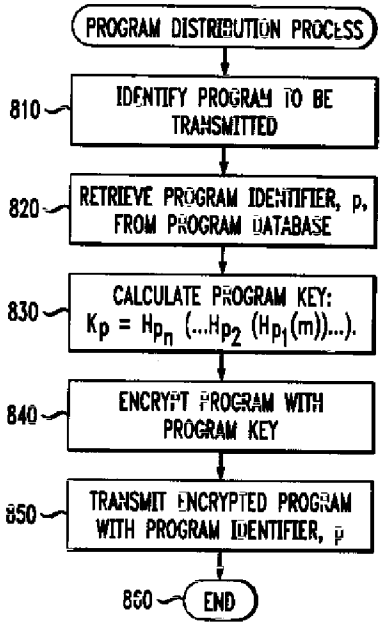
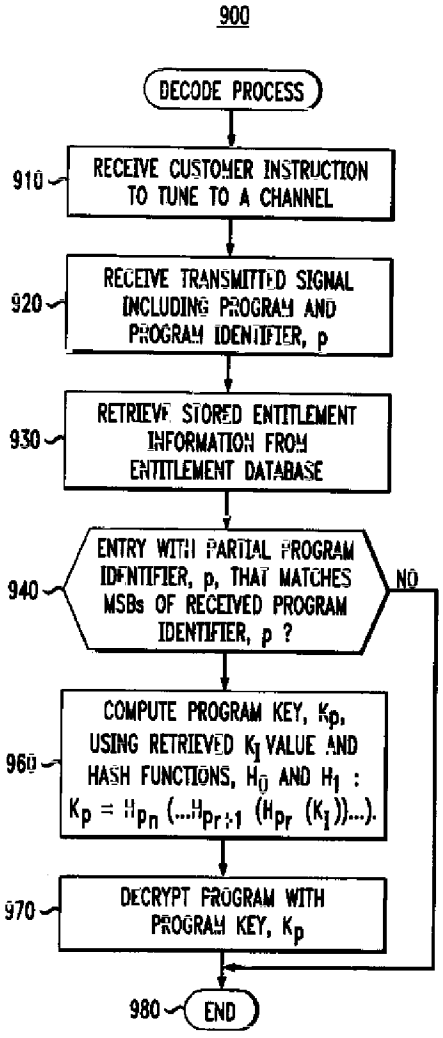

PROGRAM DISTRIBUTION PROCESS
810
IDENTIFY PROGRAM TO BE TRANSMITTED
820
RETRIEVE PROGRAM IDENTIFIER, p, FROM PROGRAM DATABASE
830
CALCULATE PROGRAM KEY: $K_p = H_{p_n}(...H_{p_2}(H_{p_1}(m))...)$.
840
ENCRYPT PROGRAM WITH PROGRAM KEY
850
TRANSMIT ENCRYPTED PROGRAM WITH PROGRAM IDENTIFIER, p
860
END

FIG. 9


900
DECODE PROCESS
910
RECEIVE CUSTOMER INSTRUCTION TO TUNE TO A CHANNEL
920
RECEIVE TRANSMITTED SIGNAL INCLUDING PROGRAM AND PROGRAM IDENTIFIER, p
930
RETRIEVE STORED ENTITLEMENT INFORMATION FROM ENTITLEMENT DATABASE
940
ENTRY WITH PARTIAL PROGRAM IDENTIFIER, p, THAT MATCHES MSBs OF RECEIVED PROGRAM IDENTIFIER, p ?
NO
960
COMPUTE PROGRAM KEY, Kp, USING RETRIEVED K1 VALUE AND HASH FUNCTIONS, H0 AND H1 : Kp = Hpn (...Hpr+1 (Hpr (K1))...).
970
DECRYPT PROGRAM WITH PROGRAM KEY, Kp
980
END

A system for restricting access to transmitted programming content is disclosed, which transmits a program identifier with the encrypted programming content. A set-top terminal or similar mechanism restricts access to the transmitted multimedia information using stored decryption keys. The set-top terminal receives entitlement information periodically from the head-end, corresponding to one or more packages of programs that the customer is entitled to for a given period. Each program is encrypted by the head-end server prior to transmission, using a program key, K_P, which may be unique to the program. The set-top terminal uses the received program identifier, p, together with the stored entitlement information, to derive the decryption key necessary to decrypt the program. Each of the *k*-bit program keys, K_P, used to encrypt transmitted programs is obtained by applying one or more pseudo-random hash functions, H, such as a length-doubling hash function, H, to a master key, *m*. The illustrative hash function, H, takes a *k*-bit binary value and produces a binary value having a length of 2*k*, with H_0 being the left half of the output of the hash function, and H_1 being the right half of the output of the hash function. A program key, K_p, is obtained by recursively applying a hash function, H_0 or H_1, to the master key, m, depending on the corresponding binary value of each bit position of the program identifier, p. The hash operation is represented in terms of an *n*-level binary tree, T, referred to as the key tree, with the master key, *m*, placed at the root of the tree. The tree is generated by applying the hash functions H_0 and H_1 to each node, until the desired number of tree levels (n) have been created. The program keys, K_p, correspond to the leaf nodes at the bottom level of the tree. The program identifier, p, associated with each program key, K_p, corresponds to the path through the key tree from the root to the desired leaf node.

2 Representative Drawing

Figure 1

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-036517
(43)Date of publication of application : 09.02.2001

(51)Int.CI. H04L 9/08
G09C 1/00
H04N 5/44
H04N 7/08
H04N 7/081
H04N 7/16
H04N 7/167

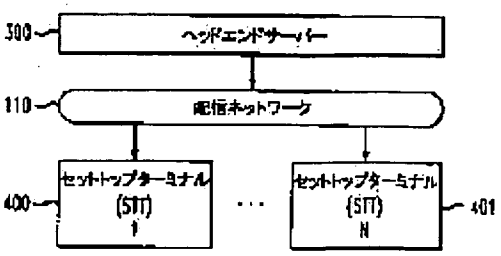
(21)Application number : 2000-135069
(22)Date of filing : 08.05.2000

(71)Applicant : LUCENT TECHNOL INC
(72)Inventor : BLEICHENBACHER DANIEL
WOOL AVISHAI

(30)Priority
Priority number : 99 307643 Priority date : 07.05.1999 Priority country : US

(54) METHOD FOR TRANSMITTING PROGRAM TO LIMIT ACCESS TO END USER AND METHOD FOR DECODING ENCRYPTED PROGRAM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a system to limit access to contents of transmission program such as television program.
SOLUTION: A transmitter or a head end server is used by a service provider to transmit encrypted programming contents to one or a plurality of customers. A program identifier (p) used to identify a program is transmitted to the customers together with programming contents. Each customer uses a set-top terminal or an interpretation key to provide a limited access to transmission multimedia information as other device. The set-top terminal 400 or the like receives entitlement information corresponding to a package of one or a plurality of programs that can normally be received for a period from a head end.



LEGAL STATUS

[Date of request for examination]	13.08.2002
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	

CLAIMS

[Claim(s)]
[Claim 1] The step which assigns the program identifier which is the approach of transmitting the program which can carry out access restriction to an end user, and has (A) binary value to said program, (B) The step which enciphers said program by using the step which defines at least one master key, and the program key obtained by applying at least one Hash Function to said master key based on the binary value of the (C) aforementioned program identifier, (D) Approach characterized by having the step which sends said enciphered program to said end user with said program identifier.
[Claim 2] Said program identifier is an approach according to claim 1 characterized by applying one of said the Hash Functions to each location of n bits of said program identifier according to the bit value to which it becomes from n bits and said program identifier corresponds.
[Claim 3] (E) The approach according to claim 1 characterized by having further the step which provides said end user with entitlement information based on the set of the program acquired by said end user.
[Claim 4] The approach according to claim 3 characterized by including some key trees based on the set of the program acquired by said end user in said entitlement information.
[Claim 5] Said end user is an approach according to claim 3 characterized by using said program identifier in order to obtain said program key from said memorized entitlement information.
[Claim 6] Said program identifier is an approach according to claim 1 characterized by interleaving with transmission of said encryption program.
[Claim 7] Said program identifier is an approach according to claim 1 characterized by being transmitted on a control channel.
[Claim 8] The approach characterized by to have the step enciphered using the program key which is the approach of transmitting a program to two or more end users, and was obtained by applying a Hash Function to the master key based on the binary value of each bit position of said program identifier for the program which has (A) program identifier recurrently, and the step which transmits the program which carried out (B) encryption, and said program identifier to said end user.
[Claim 9] Said program identifier is an approach according to claim 8 characterized by applying said Hash Function to each location of n bits of said program identifier according to the bit value to which it becomes from n bits and said program identifier corresponds.
[Claim 10] (C) The approach according to claim 8 characterized by having further the step which provides said end user with entitlement information based on the set of the program acquired by said end user.
[Claim 11] The approach according to claim 10 characterized by including some key trees based on the set of the program acquired by said end user in said entitlement information.
[Claim 12] Said end user is an approach according to claim 10 characterized by using said program identifier in order to obtain said program key from said memorized entitlement information.
[Claim 13] Said program identifier is an approach according to claim 8 characterized by interleaving with transmission of said encryption program.

[Claim 14] Said program identifier is an approach according to claim 8 characterized by being transmitted on a control channel.
[Claim 15] It is the approach of transmitting the program corresponding to at least one program package to two or more end users. (A) The step which provides said end user with entitlement information based on the set of the program acquired by said end user, (B) The step enciphered using the program key obtained by applying a Hash Function to the master key based on the binary value of each bit position of said program identifier for the program which has a program identifier recurrently, (C) Have further the step which transmits said program identifier to said end user with the enciphered program, and if said end user is a just user of said program Said end user is an approach characterized by obtaining said program key from the memorized entitlement information.
[Claim 16] Said program identifier is an approach according to claim 15 characterized by applying one of said the Hash Functions to each location of n bits of said program identifier according to the bit value to which it becomes from n bits and said program identifier corresponds.
[Claim 17] The approach according to claim 15 characterized by including some key trees based on the set of the program acquired by said end user in said entitlement information.
[Claim 18] Said end user is an approach according to claim 15 characterized by using said program identifier in order to obtain said program key from said memorized entitlement information.
[Claim 19] Said program identifier is an approach according to claim 15 characterized by interleaving with transmission of said encryption program.
[Claim 20] Said program identifier is an approach according to claim 15 characterized by being transmitted on a control channel.
[Claim 21] The step which receives the entitlement information which is the approach of decoding the enciphered program and contains at least one middle key from a key tree based on the set of the program which said customer acquired from the provider of the (A) aforementioned program, (B) The encryption program enciphered by the program key, and the step which receives a program identifier, (C) Approach characterized by having the step which obtains said program key from the part said program identifier and said key tree were remembered to be, and the step which decodes said encryption program using the (D) aforementioned program key.
[Claim 22] It is the approach according to claim 21 which said program identifier consists of n bits, and said master key is arranged on the root of said key tree, and is characterized by generating said key tree when said key tree applies a Hash Function to each node until the tree level of n is made.
[Claim 23] It is the approach of decoding the enciphered program. From the provider of the (A) aforementioned program The step which receives the entitlement information which contains at least one middle key from the key tree based on the set of the program which a customer acquires, (B) The encryption program enciphered by the program key, and the step which receives a program identifier, (C) The step which obtains said program key from the part the key tree was remembered to be from said program identifier and said middle key by applying a Hash Function to said middle key recurrently based on the binary value of said program identifier, (D) Approach characterized by having the step which decodes said encryption program using said program key.
[Claim 24] It is the approach according to claim 23 which said program identifier consists of n bits, and said middle key corresponds to the intermediate node in the level r of said key tree, and is characterized by carrying out n-r time application of said Hash Function at said middle key.
[Claim 25] The memory which is the system which transmits the program which restricts access to an end user, and memorizes the (A) master key and a computer readout possible code, (B) It has the processor connected with said memory in actuation. This processor (a) The program identifier which has a binary value is assigned to said program. (b) Define at least one master key and said program is enciphered using a program key by applying at least one Hash Function to said master key based on the binary value of the (c)

aforementioned program identifier. (d) System characterized by constituting so that an encryption program may be transmitted to said end user with said program identifier.
[Claim 26] The memory which is the system which transmits the program to which access to an end user was restricted, and memorizes the (A) master key and the code which can be computer read, (B) It has the processor connected with said memory on actuation. Said processor (a) The program key obtained by applying a Hash Function to a master key recurrently based on the binary value of each bit position of said program identifier is used. The system characterized by constituting so that this program that enciphered this program that has a program identifier and was enciphered by the (b) aforementioned end user, and said program identifier may be transmitted.
[Claim 27] The memory which is the system which decodes the enciphered program and memorizes the (A) master key and the code which can be computer read, (B) It has the processor connected with said memory on actuation. Said processor (a) The entitlement information containing the part of the key tree based on the set of the program acquired by said customer is received from the provider of this program. (b) The encryption program enciphered by the program key and a program identifier are received. (c) System characterized by obtaining said program key from said part said program identifier and said key tree were remembered to be, and constituting so that said encryption program may be decoded using the (d) aforementioned program key.
[Claim 28] It is the medium by which the code means which can be computer read was mounted and which can be computer read. This means that can be computer read assigns the program identifier which has (a) binary value at the time of actuation to a program. (b) Define at least one master key and the program key obtained by applying at least one Hash Function to said master key based on the binary value of the (c) aforementioned program identifier is used. The medium which is characterized by transmitting this program that enciphered this program and was enciphered with the (d) aforementioned program identifier to an end user and which can be computer read.
[Claim 29] It is the medium by which the code means which can be computer read was mounted and which can be computer read. This means that can be computer read receives the entitlement information containing the part of the key tree based on the set of the program acquired by the (a) aforementioned customer at the time of actuation from the provider of this program. (b) The encryption program enciphered by the program key and a program identifier are received. (c) Medium which is characterized by obtaining said program key from said part said program identifier and said key tree were remembered to be, and decoding said encryption program using the (d) aforementioned program key and which can be computer read.

[Translation done.]

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention relates to the system which transmits the program decoded with the memorized entitlement information using the program identifier used by the set top terminal, in order to obtain a decode key required to decode a program especially about the system which restricts access to the contents of transmitting programming.
[0002]
[Description of the Prior Art] It is still more important that a service provider like a cable television operator or a digital satellite service operator offers the package of the channel to which a majority of a television viewer's population is satisfied, or a program as the number of channels with an available television viewer increases and the range of the available contents of programming increases in number by such channel. Generally development of the package with which a customer is provided is a marketing function. A service provider is wanted to offer the package of various sizes generally. For example, they are all programs, the combination between them, etc. from one program.
[0003] A service provider usually broadcasts a television program from the transmitter called a "head end" to many customers. Each customer is usually concerned with a part of programming to receive. For example, in a broadcast environment, any man can receive programming transmitted with a suitable receiver like an antenna or a satellite disk. In order to restrict access of a program only to the normal customer who purchased the package, a service provider usually enciphers a transmitting program and contains 1 or two or more code machines in a customer. A set top terminal (STT) is offered. By such approach, a set top terminal receives encryption transmission and the program which a customer looks at is enciphered. Nothing is carried out but this.
[0004] In order that the confidentiality memorized in the set top terminal may make piracy of high information min, a set top terminal is usually equipped with a secure processor or secure memory. This secure memory has the capacity of several kilobits order, and memorizes a code key. Generally secure memory is not volatility but tamper REJISUTANTO. Moreover, secure memory has that it can write [much] in and can carry out the repro gram of the key for every accounting period. Since the secure memory capacity of the conventional set top terminal is restricted, the number of the keys memorized will be restricted and the number of the packages which a service provider offers will also be restricted. The number of the programs which a service provider broadcasts to the accounting period of a moon unit may usually be the order of 200,000.
[0005] The conventional set top terminal has a thing containing bit VEKUTORU which has a bit entry corresponding to each package of the program which a service provider offers. If a specific customer is the normal addressee of a package, the bit entry in the bit vector memorized in a set top terminal will be set to "1." After that, all the programs that a service provider transmits are enciphered by one key. If a program is received, a set top terminal will judge whether the bit entry which accesses and corresponds to a bit vector is set. If the bit entry is set, as for a set top terminal, a program will be decoded using one memorized code

machine.
[0006] Although it seems to a theory top that flexibility is attained by the bit vector method by offering one bit entry to each package (a package consisting of one program generally), the die length of a bit vector is not practical in the system which transmits many programs to one accounting period. Moreover, the access control in such a system is exclusively given by the entry in a bit vector, and is not code-like (cryptographic). Therefore, if a customer can write in a bit vector and can set all bits to "1", a customer will be able to access all programs.
[0007] Moreover, a program is divided into each package and there are some as which all the programs in a package are enciphered using the same key. Each package corresponds to one television channel. A set top terminal memorizes the decode key to each package the customer of whose is a normal addressee. Therefore, if a program is included in two or more packages, that program must be broadcast again for corresponding each package of every, and will be enciphered in this the transmission of each by the code key corresponding to a specific package. Although it is cryptography-like [an access control], by the overhead about broadcasting programming again repeatedly, it will not be realistic, and will carry out arranging the same program as much packages, and flexibility will be restricted in the design of the package of a program.
[0008] although the conventional system which encipher such contents of a program and be transmit be comparatively successful about restrict access only to a normal customer , it have not make it possible to provide a customer with the package with which a large number which include much programs , without make an overhead increase fairly differ , without a service provider like a television network exceed the secure memory capacity to which the set top terminal be restricted . The cryptography-approach and equipment which restrict access to the contents of transmitting programming to the "Vspace system" indicated by the United States patent applications 08/912186 (August 15, 1997 application) are indicated.
[0009] Each program in a Vspace system is enciphered by the head end server before transmission using the program key kP. Each program key is the linearity combination of the set with which the master key M was defined. The program identifier which identifies a program is transmitted with the contents of encryption programming. A customer's set top terminal can obtain a decode key only from the entitlement information recorded on the program identifier p which received, and the front. A Vspace system offers a cryptography-access-control mechanism, enabling the package which is supple, without extending a program header fairly (only a program identifier being transmitted with a program). It is because it is not necessary to broadcast a program again for corresponding each package of every.
[0010]
[Means for Solving the Problem] Generally, the contents of programming enciphered by 1 or two or more customers by the service provider using the transmitter thru/or the head end server are transmitted. The program identifier p used for identifying a program is transmitted to a customer with the contents of programming. Each customer has other devices in which access restricted to transmitting multimedia information using the set top terminal thru/or the decode key is given. A set top terminal receives 1 which can receive to normal at a period with a customer, or the entitlement information corresponding to the package of two or more programs from a head end.
[0011] Each program is enciphered by the head end server before transmission using the program key kp. the program key kp of an individual -- the program -- unique -- making . In addition to transmission of the enciphered program, a head end server transmits the program identifier p to a set top terminal. A set top terminal obtains a decode key required to decode a program using the program identifier p which received with the memorized entitlement information. In this approach, if a customer is the normal user of a specific program, a set top terminal can obtain the program key kp enciphered using the information memorized and received, and can decode the program enciphered using that program key kp after that. In an example, the program identifier p can be interleaved to a part of program, and can be transmitted on a separate

exclusive control channel.
[0012] Each of k-bit program key kp used for enciphering a transmitting program can be obtained by applying 1 or two or more pseudo-random Hash Functions to a master key m. As an example, Hash Function H which doubles die length can be used. Therefore, Hash Function H takes a k bit binary value, and makes the binary value of the die length of 2k. The output of Hash Function H can be expressed as pair H0 of k-bit binary value as H1. Here, H0 can be identified as a left half of the output of the Hash Function concerned, and H1 can be identified as a right half of the output of the Hash Function concerned.
[0013] As an example, the program key kp can be obtained according to the binary value to which each bit position of the program identifier p corresponds by applying recurrently Hash Functions H0 or H1 to a master key. Therefore, if the program identifier p consists of m bits, one side of Hash Functions H0 or H1 will be applied to each bit position of n of the program identifier p according to the bit value to which the program identifier p corresponds. First, one side of Hash Functions H0 or H1 is applied to a master key according to the binary value which is the leftmost digit bit of the program identifier p. After that, according to the binary value of a corresponding bit, one side of Hash Functions H0 or H1 is applied to the result of a pre- hash operation to each remaining bit position (n-1). Count of the program key kp can be expressed as follows.
[Equation 1]

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

[0014] Such a hash operation can be expressed in relation to n level binary tree T (called a key tree) by which the root 2 master key m of a tree is arranged. A tree is generable by applying Hash Functions H0 and H1 to each node until a desired number of tree-level (n) is made. The program key kp corresponds to the leaf (leaf) node in the bottom (bottom) level of a tree. The binary index (the same the program identifier [And] p) corresponding to each program key kp corresponds to the pass (way) which passes along the key tree from the root to a desired leaf node. Therefore, the index thru/or label of Node u is connection of the label on H on the pass from the root to Node u. T (u) can calculate any key of the program in subtree T (u) by carrying out time (n-r) actuation of the Hash Function to the internal node u (u1, ..., ur) in depth r in the subtree which makes Node u the root, i.e., the key tree which has the partial program identifier p showing the set of the program identifier p corresponding to the leaf in the subtree of Node u.
[0015]
[Embodiment of the Invention] Drawing 1 has shown the network environment which transmits video, an audio, and encryption multimedia information like data to 1 or two or more customers who have the set top terminals 400-401 through 1 or two or more distribution networks 110 using a transmitter like the head end server 300 from a service provider. This head end server 300 argues in relation to drawing 3 in the bottom, and argues about the set top terminal 400 in relation to drawing 4 in the bottom. In this specification, a set top terminal includes any device in which access restriction is given to the multimedia information transmitted using the decode key. For example, a computer configuration and a communication link device are included. A service provider may download the software which a set top terminal performs. A network 110 can be made into the wireless broadcasting network which distributes contents of programming like digital satellite service (DSSTM), a cable television network (CATV), a public switching network (PSTN), an optical network, ISDN, and a cable network like the Internet.
[0016] The set top terminal 400 receives entitlement information intermittently from the head end server 300, and enables a customer to access the program whose customer is a registered user between a certain time intervals (for example, accounting period). In this specification, a package is the set of a predetermined program and a certain program can belong to 1 or two or more packages. A program means all of continuous multimedia transmission of the episode of television, or specific die length like a movie. Entitlement information is downloadable in the set top terminal 400 from the head end server 300 using

which suitable secure one way or bidirectional protocol.
[0017] Program key and program identifier each transmitting program is enciphered by the head end server 300 using the program key kp. This program key kp can be made unique to a program. Suitable encryption and a security technique are indicated by reference, B.Schneier, and Applied Cryptography (2d ed.1997). In addition to transmission of an encryption program, the head end server 300 also transmits a n bit program identifier to the set top terminal 400. This is used by the set top terminal 400 with the memorized entitled information, and as shown in a detail, it obtains a decode key required to decode a program in the bottom.
[0018] The program identifier p is not chosen as arbitration so that the item of the bottom entitled assignment of the program identifier to a program may explain. In a desirable example, the program identifier p can consist of the 32-bit value transmitted in the ECM field specified to MPEG-2 criterion. In this case, if it is the registered user of the program of specification [a customer], the set top terminal 400 can obtain the program key kp from the information memorized and received, and it can use the program key kp so that an encryption program may be decoded after that.
[0019] According to the further description of this invention, each of the k-bit program key kp used for an encryption transmitting program can be obtained by applying 1 or two or more pseudo-random Hash Functions to a master key m. Explanation of a suitable pseudo-random Hash Function is indicated by reference and O.Goldreich et al. and "How to Construct Random Functions" J.ACM and 33:792-807 (1986).
[0020] As an example, it is secure in cryptography, and the Hash Function which doubles die length is used as follows.
H: {0, 1} k->{0, 1}2k -- here, k is the die length of the program key kp. Therefore, Hash Function H takes the binary value of k bits, and makes the binary value of die-length 2k. The output of this Hash Function H can be expressed as pair H0 of a k bit binary value as H1. Here, H0 is the left-hand side one half (left-hand side digit bit) of the output of Hash Function H, and is H. {1} is the right-hand side one half (right-hand side digit bit) of the output of Hash Function H. H0 and H1 can be called a separate Hash Function.
[0021] If it is k= 160, H can be specified using secret hash standard SHA-1 which is indicated by reference, Secure Hash Standard, National Institute of Standards and Technology, NIST FIPS PUB 180-1, and U.S.Dept.of Commerce (April, 1995). That is, H0 is set to SHA-1 (x||0), and H1 turns into SHA-1 (x||1). Here, 0 and 1 are the bit strings of all the bit strings 1 of 0 altogether, respectively.
[0022] The program key kp can be obtained by applying recurrently 1 or two or more Hash Functions to a master key m according to the binary value of the program identifier p. As an example, the program key kp can be obtained by applying recurrently one side of Hash Functions H0 or H1 to a master key m according to the binary value of each bit position of the program identifier p. Generally, if the program identifier p consists of n bits, according to the bit value to which the program identifier p corresponds, one side of Hash Functions H0 or H1 will be applied to each of the bit position of n of the program identifier p (it starts from a leftmost bit).
[0023] One side of Hash Functions H0 or H1 is first applied to a master key according to the binary value which is a leftmost digit bit. After that, according to the binary value which is the bit to which one side of Hash Functions H0 or H1 corresponds, it is applied to the result of pre- hash actuation to each remaining bit position (n-1). This hash actuation can be expressed as follows so that the item of a title called lower "key tree" may explain.
[Equation 2]

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

[0024] As mentioned above, the head end server 300 transmits the program identifier p with an encryption program. Therefore, if the program identifier p is given, the set top terminal 400 must obtain the program key kp used for decode of a receiving agent. As mentioned above, the program key kp can be obtained by applying recurrently 1 or two or more Hash Functions to a master key m according to the binary value of

the program identifier p. The program key kp must be obtained by a customer's set top terminal 400, using indirectly the memorized entitlement information and the program identifier p which received which is explained in the bottom.

[0025] As explained on the key tree, the program key kp can be obtained by using recurrently 1 or two or more Hash Functions for a master key m according to the binary value of the program identifier p. The k-bit single master key m is used. The bit of the program identifier p can be expressed as p= (p1, ..., pn). Here, p1 is a leftmost digit bit and is a rightmost digit bit. The cryptographic key kp to the program which has the program identifier p can be defined as follows.

[Equation 3]

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

[0026] Hash actuation can be expressed as a perfect n level binary tree T like the key tree 200 shown in drawing 2 . The key tree 200 shown in drawing 2 corresponds to the example of mounting which has the program identifier p which consists of a triplet. As shown in drawing 2 , a master key m is arranged on the root 210 of a tree 200. The program key kp corresponds to a leaf node like leaf nodes 240-247. The index corresponding to each program key kp shown in drawing 2 like the index 011 corresponding to the program key kp of the DERIFU node 243 shows the pass which lets the key tree 200 from the root 210 to a leaf node 243 pass. For example, the program key kp of 243 can be obtained by following with the left edge (H0) from the root 210, the right edge (H1) from a node 220, and the right edge (H1) from a node 232. That is, H1 is further applied for H0 to the 2nd hash result. The program key kp011 can be obtained.

[0027] Therefore, the label of a node u like a node 243 is what connected the label on the edge of the pass to Node u from the root 210. The label of each node can be specified by the program identifier p. Since the subtree which makes Node u the root is expressed, T (u) is used (namely, since the set of the program identifier p corresponding to the leaf in the subtree of Node u is expressed). The internal node u in depth r in the key tree 200 has the partial program identifier p (u1, ..., ur), and can calculate the key of which program in subtree T (u) to these. Any key of the program in the subtree of Node u is calculable by carrying out time (n-r) actuation of the Hash Function. Specifically, it uses so that the value of each bit of the low digit of (n-r) of the program identifier p may direct suitable Hash Functions H0 or H1. Therefore, the program key kp corresponding to Node u can function as an entitlement to all the programs in the subtree of Node u.

[0028] If Function H is a pseudo-random generator, mapping kp{0, 1} ->[n] {0, 1} k of the program key which the master key m parameterized is a pseudo-random function. This is indicated by reference, and O.Goldreich et al. and "How toConstruct Random Functions" J.ACM and 33:792-807 (1986).

[0029] System component drawing 3 is the block diagram showing the head end server's 300 AKI theque char. A head end shall be related with the service provider of the arbitration which transmits a television network, a cable employment person, a digital satellite service employment person, or the contents of encryption programming. the head end server 300 -- for example, IBM -- it can mount with RS6000 server which Corp(s) and manufactures, and the function and actuation of this invention can be performed. The head end server 300 is equipped with related memory like a processor 310 and the data storage device 320. A processor 310 may be mounted as a single processor and may be mounted as some processors which operate to juxtaposition. The data storage device 320 and ROM are made to memorize 1 or two or more instructions, and a processor 310 enables it to perform by taking out and interpreting.

[0030] As mentioned above, the data storage device 320 is equipped with the master key database 350 which memorizes a master key m. For example, a master key m can be updated like [for every accounting period]. Moreover, the data storage device 320 has the program database 500 so that it may explain in relation to drawing 5 in the bottom. The program database 500 presents the program identifier p and the related package corresponding to each program. moreover, drawing 7 R> -- the data storage device 320 has the

entitlement information delivery process 700 and the program delivery process 800 so that it may explain in relation to 7 and 8.

[0031] Generally, the entitlement information delivery process 700 generates and distributes the entitlement information which each customer needs to accessing the program which is a registered user. Moreover, the program delivery process 800 obtains the program key kp based on the program identifier p assigned to the program, in order to encipher a program and to transmit by the program identifier p.

[0032] The communication link port 330 links the head end server 300 to each connected receiver like the set top terminal 400 which showed the head end server 300 to the network 110 at a bond and drawing 1 .

[0033] Drawing 4 is the block diagram showing the AKI theque char of the set top terminal 400. The set top terminal 400 can be mounted as a set top terminal (STT) corresponding to television, and it can be changed so that the function and actuation of this invention may be performed. The set top terminal 400 is equipped with a processor 410 and memory like data storage 420, and the communication link port 430, and operates by the same approach as the above hardware relevant to drawing 3 .

[0034] Data storage 420 is equipped with the entitlement database 600 memorizable into the secure part of data storage 420 so that it may explain in relation to drawing 6 in the bottom. The entitlement database 600 contains the part of the key tree 200 required in order that a customer may get the program key kp to the program which has an entitlement. Moreover, data storage 420 is equipped with Hash Functions H0 and H1 (440). Moreover, data storage 420 includes the decoding process 900 so that it may explain in relation to drawing 9 in the bottom. Generally, using the program identifier p received in order to obtain the program key kp, and the memorized entitlement information 600, in order to decode a program, the program key kp is used for the decoding process 900, and it decodes the program whose customer has an entitlement.

[0035] Drawing 5 shows the program database 500 which memorizes information on each program p transmitted by the head end server 300. This information is transmitted to for example, an accounting period with the program identifier p to which that program belongs and which packs and corresponds. The program database 500 holds two or more decodings like records 505-520. These are related with a different program, respectively. The program database 500 contains the program identifier p which corresponds in the field 535 including directions of the corresponding package with which the program belongs in the field 530 to each program identifier identified by the program name in the field 525.

[0036] Drawing 6 shows the entitlement database 600 containing the part of the key tree 200 required for a customer to get the program key kp to the program which has an entitlement. As mentioned above, T (u) expresses the set of the program identifier p corresponding to the leaf nodes 240-247 in the subtree which makes Node u the root, i.e., the subtree of Node u. For example, supposing a customer has an entitlement about receiving four programs corresponding to leaf nodes 240-243, entitlement information will consist of a middle key corresponding to a node 220. In this approach, if needed, suitable Hash Functions H0 and H1 (440) can be used in order to obtain the program key kp to each nodes 230, 232, 240-243 in the subtree of a node 220.

[0037] The entitlement database 600 shown by drawing 6 is a registered user who receives four programs corresponding to leaf nodes 240-243 (there is an entitlement), and is a registered user who receives two programs corresponding to leaf nodes 246-247. Therefore, the entitlement information recorded on the entitlement database 600 consists of a middle key corresponding to a node 220 and a node 236. nodes 220 and 236 -- it is alike, respectively, and it receives, and the entitlement information recorded on the entitlement database 600 has the middle key values kio and ki11, respectively, and has corresponding directions of the partial program identifier p. The approach by which the entitlement database 600 is generated by the entitlement information delivery process 700 based on the package of the program which the customer chose is explained in relation to drawing 7 in the bottom.

[0038] A small entitlement is establishable to the set of many programs of various sizes using the tree method of program packaging this invention. The target set S is established using the set of the program

packed. The minimum set of a tree node with which a subtree covers the target set S correctly is obtained as follows.

[Equation 4]

$T(S)=Z\subseteq T$ ただし、$\bigcup_{u\in Z} T(u)=S$ 、かつ、$|Z|$ は最小であるように

[0039] The entitlement information over Package S is the set ki of the middle key held in the node of T (S). As shown in a top, the set top terminal 400 decodes the program in S (accepting it) correctly with the set of this key. Theoretically, the tree method of this invention can build the entitlement information over the target set S of which arbitration. furthermore -- however, if the program identifier p is assigned to arbitration, entitlement information will become so large that it is not allowed for the secure memory to which the set top terminal 400 was restricted.

[0040] a process -- as mentioned above, the head end server 300 performs the entitlement information delivery process 700 shown in drawing 7 , and generates and distributes the entitlement database 600 required for each user in order to access the program which is a registered user. As mentioned above, the entitlement database 600 consists of corresponding directions and the corresponding middle key value ki of a partial program identifier to each node of the key tree 200 required for a customer to get the program key kp to the program which is a registered user.

[0041] Therefore, the entitlement information delivery process 700 identifies first the program which the customer chose (710). After that, the entitlement information delivery process 700 finds minimum set [of a tree node] T (S). The subtree covers the target set S correctly. The target set S is disassembled to the maximum De Dis joint interval of the KONSEKYUTIBU program identifier p (720). Two program identifiers p are considered to be KONSEKYUTIBU when the integer over the binary expression is KONSEKYUTIBU.

[0042] And covering T (S) is found to each interval (730). The corresponding partial program identifier p held in the node of covering T (S) to Set ki and each interval of a middle key is generated (740). At the end, the generated entitlement information downloads to the set top terminal 400 with the head end server 300 (750), and program control is completed (760).

[0043] The number of the intervals in the target set S can be set to I (S). In order to calculate covering T (S) to the single interval of the program identifier p to the order of the tree node of n, the key tree 200 of depth n must be asked. Therefore, the time amount complexity of the entitlement information delivery process 700 serves as order of I(S) -n. Similarly, the magnitude of minimum covering T (S) serves as order of I(S) -n. The program identifier p which enables the program of related contents to carry out packaging of them efficiently should be assigned. In an example, a fundamental package is the gestalt of all the program identifiers p that have the bit prefix mu.

[0044] The entitlement of such a single topic package is a single key in the key tree 200. Moreover, a multi-topic package can be assembled without a side effect. Entitlement information is only the set of a key to each TOPICS which consists of a multi-TOPICS package. According to this invention, the package specified by Prefix mu does not force to the set top terminal 400 so that a program may be decoded using zero prefix of the same die length.

[0045] As mentioned above, the head end server 300 performs the program delivery process 800 shown in drawing 8 , and in order to decode a program and to transmit using the program identifier p, he gets the program key kp based on the program identifier p assigned to the program and the master key m. The program delivery process 800 is important for performing in off-line thru/or the real time except an actual transmitting step. As shown in drawing 8 , the program delivery process 800 starts the process using the principle of this invention by identifying the program which should be transmitted (810).

[0046] After that, the program delivery process 800 takes out the program identifier p corresponding to the program from the program database 500 (820), and calculates the program key kp corresponding to the

program (830). And a program is enciphered using the program key kp calculated at the front step (840). Finally, the program delivery process 800 transmits the program enciphered with the program identifier p (850), and program control ends it (860).

[0047] It is important to suppose that it is possible to obtain the program key kp required for the program identifier p to be interleaved periodically, able to transmit it through transmission of program information, and for a customer change a channel at the time of a program, and decode a program. In another example, the program identifier p can be continuously transmitted on another control channel like a Barker channel.

[0048] As mentioned above, the set top terminal 400 performs the decoding process 900 shown in drawing 9 , using the entitlement information 600 and the received program identifier p memorized in order to obtain the program key kp, in order to decode the program, the program key kp is used and a customer decodes the program by which the entitlement is carried out. As shown in drawing 9 , the decoding process 900 starts the process which used the principle of this invention on the occasion of the reception of the customer directions made to tune up to a specific channel (910).

[0049] After that, the set top terminal 400 receives the suitable signal containing the enciphered program identifier p which was programmed and transmitted (920). The decoding process 900 takes out the entitlement information memorized from the entitlement database 600 (930). It judges whether the transmitted program is included (940). When the entry which has the partial-program identifier p which agrees in the leftmost digit bit of the receiving-agent identifier p at step 940 is judged not to exist in the entitlement database 600, a customer does not have an entitlement to the selected program and program control is ended (980).

[0050] However, if an entry exists in the entitlement database 600 which has the partial-program identifier p corresponding to the leftmost digit bit of the received program identifier p, a customer has an entitlement to the selected program. Therefore, the program key kp is calculated using the middle key ki taken out from the entry of the entitlement database 600 (960). Specifically, the program key kp is calculated by operating suitable Hash Functions H0 or H1 so that each value of the bit of the low (n-r) order of the program identifier p may direct as follows.

[Equation 5]

$$K_p = H_{p_n}(\ldots H_{p_{r+1}}(H_{p_r}(K_I))\ldots)$$

[0051] Finally, the program is decoded using the obtained program key kp (970), and ends program control (980). When the received program is not a part of a customer's entitlement here, it is important that there is no entitlement information which has the partial identifier p corresponding to the low bit of the program identifier p which received with the transmitting program in the entitlement database 600.

[0052] The decoding process 900 obtains a decode key, or moreover, as mentioned above Before a customer judges whether there is any entitlement to a demand channel In order that it can wait for a customer to demand a specific channel and the decoding process 900 may obtain the transmitting program identifier p instead, all channels are scanned periodically. It is important that the decode key to the storage in data storage 420 can be obtained, and a customer's entitlement can be judged beforehand again.

[0053] a suitable Hash Function -- as mentioned above, if Hash Function H is a pseudo-random bit generation machine, it can prove that mapping of p->kp is a pseudo-random function. Therefore, a code key cannot be predicted if actual Hash Function H is strong in cryptography. Therefore, if a piracy person has access only to encryption program broadcasting, it will not be able to break through a code in the knowledge about the key generated using the tree method of this invention. Therefore, only one concerns only become ensuring that video encryption algorithm can oppose to a well-known plain text attack.

[0054] Hash Function H should hold two properties. Calculating Input x has that it must be difficult noting that the one half H0 of an image (x) or H1 (x) is given to the 1st to Hash Function H. Though this knows the image of both these one half, it is actually materialized also to the cryptography-hash [which] H with it

difficult [to carry out an inverted arch]. Moreover, though H1 (x) was known, it must be difficult to calculate H0 (x), and the reverse of a thing is also the same. Even if it is difficult fundamentally to carry out the inverted arch of the function H, when the key of one one half is known, it becomes easier to complete the key of the remaining one half. If that is right, the piracy person who knows Program kp to Node u can calculate the key to the SHIBURINGU (sibling: sibling) node v, and can calculate the key to all the programs in the subtree of Node v.

[0055] As one advantage of the tree method according to this invention, merge of an entitlement carried out in piracy may be made in inefficient. Pair p1, p2, and those ********** of a SHIBURINGU program are considered. A piracy person assumes that the program key kp corresponding to the programs p1 and p2 of both which are two one half of H (kp (u)) is known. A piracy person still cannot do the inverted arch of the H, and cannot calculate kp (u). It is because H is a cryptography-Hash Function. Therefore, the entitlement carried out in the merged piracy must contain both kp (p1) and kp (p2) instead of compact kp (u). therefore, it is not a strategy good for a piracy person to divide to two or more set top terminals 400 which use a CHIPU (it is -- although -- it differs) entitlement. It is because a union ****** entitlement becomes very large.

[0056] As mentioned above, the suitable pseudo-random Hash Function is indicated by reference, and O.Goldreich et al. and "How to Construct Random Functions" J.ACM and 33:792-807 (1986).

[Translation done.]

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] This invention relates to the system which transmits the program decoded with the memorized entitlement information using the program identifier used by the set top terminal, in order to obtain a decode key required to decode a program especially about the system which restricts access to the contents of transmitting programming.

[Translation done.]

PRIOR ART

[Description of the Prior Art] It is still more important that a service provider like a cable television operator or a digital satellite service operator offers the package of the channel to which a majority of a television viewer's population is satisfied, or a program as the number of channels with an available television viewer increases and the range of the available contents of programming increases in number by such channel. Generally development of the package with which a customer is provided is a marketing function. A service provider is wanted to offer the package of various sizes generally. For example, they are all programs, the combination between them, etc. from one program.

[0003] A service provider usually broadcasts a television program from the transmitter called a "head end" to many customers. Each customer is usually concerned with a part of programming to receive. For example, in a broadcast environment, any man can receive programming transmitted with a suitable receiver like an antenna or a satellite disk. In order to restrict access of a program only to the normal customer who purchased the package, a service provider usually enciphers a transmitting program and contains 1 or two or more code machines in a customer. A set top terminal (STT) is offered. By such approach, a set top terminal receives encryption transmission and the program which a customer looks at is enciphered. Nothing is carried out but this.

[0004] In order that the confidentiality memorized in the set top terminal may make piracy of high information min, a set top terminal is usually equipped with a secure processor or secure memory. This secure memory has the capacity of several kilobits order, and memorizes a code key. Generally secure memory is not volatility but tamper REJISUTANTO. Moreover, secure memory has that it can write [much] in and can carry out the repro gram of the key for every accounting period. Since the secure memory capacity of the conventional set top terminal is restricted, the number of the keys memorized will be restricted and the number of the packages which a service provider offers will also be restricted. The number of the programs which a service provider broadcasts to the accounting period of a moon unit may usually be the order of 200,000.

[0005] The conventional set top terminal has a thing containing bit VEKUTORU which has a bit entry corresponding to each package of the program which a service provider offers. If a specific customer is the normal addressee of a package, the bit entry in the bit vector memorized in a set top terminal will be set to "1." After that, all the programs that a service provider transmits are enciphered by one key. If a program is received, a set top terminal will judge whether the bit entry which accesses and corresponds to a bit vector is set. If the bit entry is set, as for a set top terminal, a program will be decoded using one memorized code machine.

[0006] Although it seems to a theory top that flexibility is attained by the bit vector method by offering one bit entry to each package (a package consisting of one program generally), the die length of a bit vector is not practical in the system which transmits many programs to one accounting period. Moreover, the access control in such a system is exclusively given by the entry in a bit vector, and is not code-like (cryptographic). Therefore, if a customer can write in a bit vector and can set all bits to "1", a customer will be able to access all programs.

[0007] Moreover, a program is divided into each package and there are some as which all the programs in a package are enciphered using the same key. Each package corresponds to one television channel. A set top terminal memorizes the decode key to each package the customer of whose is a normal addressee. Therefore, if a program is included in two or more packages, that program must be broadcast again for corresponding each package of every, and will be enciphered in this the transmission of each by the code key corresponding to a specific package. Although it is cryptography-like [an access control], by the overhead about broadcasting programming again repeatedly, it will not be realistic, and will carry out arranging the same program as much packages, and flexibility will be restricted in the design of the package of a program.

[0008] although the conventional system which encipher such contents of a program and be transmit be comparatively successful about restrict access only to a normal customer , it have not make it possible to provide a customer with the package with which a large number which include much programs , without make an overhead increase fairly differ , without a service provider like a television network exceed the secure memory capacity to which the set top terminal be restricted . The cryptography-approach and equipment which restrict access to the contents of transmitting programming to the "Vspace system" indicated by the United States patent applications 08/912186 (August 15, 1997 application) are indicated.

[0009] Each program in a Vspace system is enciphered by the head end server before transmission using the program key kP. Each program key is the linearity combination of the set with which the master key M was defined. The program identifier which identifies a program is transmitted with the contents of encryption programming. A customer's set top terminal can obtain a decode key only from the entitlement information recorded on the program identifier p which received, and the front. A Vspace system offers a cryptography-access-control mechanism, enabling the package which is supple, without extending a program header fairly (only a program identifier being transmitted with a program). It is because it is not necessary to broadcast a program again for corresponding each package of every.

[Translation done.]

MEANS

[Means for Solving the Problem] Generally, the contents of programming enciphered by 1 or two or more customers by the service provider using the transmitter thru/or the head end server are transmitted. The program identifier p used for identifying a program is transmitted to a customer with the contents of programming. Each customer has other devices in which access restricted to transmitting multimedia information using the set top terminal thru/or the decode key is given. A set top terminal receives 1 which can receive to normal at a period with a customer, or the entitlement information corresponding to the package of two or more programs from a head end.

[0011] Each program is enciphered by the head end server before transmission using the program key kp. the program key kp of an individual -- the program -- unique -- making . In addition to transmission of the enciphered program, a head end server transmits the program identifier p to a set top terminal. A set top terminal obtains a decode key required to decode a program using the program identifier p which received with the memorized entitlement information. In this approach, if a customer is the normal user of a specific program, a set top terminal can obtain the program key kp enciphered using the information memorized and received, and can decode the program enciphered using that program key kp after that. In an example, the program identifier p can be interleaved to a part of program, and can be transmitted on a separate exclusive control channel.

[0012] Each of k-bit program key kp used for enciphering a transmitting program can be obtained by applying 1 or two or more pseudo-random Hash Functions to a master key m. As an example, Hash Function H which doubles die length can be used. Therefore, Hash Function H takes a k bit binary value, and makes the binary value of the die length of 2k. The output of Hash Function H can be expressed as pair H0 of k-bit binary value as H1. Here, H0 can be identified as a left half of the output of the Hash Function concerned, and H1 can be identified as a right half of the output of the Hash Function concerned.

[0013] As an example, the program key kp can be obtained according to the binary value to which each bit position of the program identifier p corresponds by applying recurrently Hash Functions H0 or H1 to a master key. Therefore, if the program identifier p consists of m bits, one side of Hash Functions H0 or H1 will be applied to each bit position of n of the program identifier p according to the bit value to which the program identifier p corresponds. First, one side of Hash Functions H0 or H1 is applied to a master key according to the binary value which is the leftmost digit bit of the program identifier p. After that, according to the binary value of a corresponding bit, one side of Hash Functions H0 or H1 is applied to the result of a pre- hash operation to each remaining bit position (n-1). Count of the program key kp can be expressed as follows.

[Equation 1]

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

[0014] Such a hash operation can be expressed in relation to n level binary tree T (called a key tree) by which the root 2 master key m of a tree is arranged. A tree is generable by applying Hash Functions H0 and H1 to

each node until a desired number of tree-level (n) is made. The program key kp corresponds to the leaf (leaf) node in the bottom (bottom) level of a tree. The binary index (the same the program identifier [And] p) corresponding to each program key kp corresponds to the pass (way) which passes along the key tree from the root to a desired leaf node. Therefore, the index thru/or label of Node u is connection of the label on H on the pass from the root to Node u. T (u) can calculate any key of the program in subtree T (u) by carrying out time (n-r) actuation of the Hash Function to the internal node u (u1, ..., ur) in depth r in the subtree which makes Node u the root, i.e., the key tree which has the partial program identifier p showing the set of the program identifier p corresponding to the leaf in the subtree of Node u.
[0015]
[Embodiment of the Invention] Drawing 1 has shown the network environment which transmits video, an audio, and encryption multimedia information like data to 1 or two or more customers who have the set top terminals 400-401 through 1 or two or more distribution networks 110 using a transmitter like the head end server 300 from a service provider. This head end server 300 argues in relation to drawing 3 in the bottom, and argues about the set top terminal 400 in relation to drawing 4 in the bottom. In this specification, a set top terminal includes any device in which access restriction is given to the multimedia information transmitted using the decode key. For example, a computer configuration and a communication link device are included. A service provider may download the software which a set top terminal performs. A network 110 can be made into the wireless broadcasting network which distributes contents of programming like digital satellite service (DSSTM), a cable television network (CATV), a public switching network (PSTN), an optical network, ISDN, and a cable network like the Internet.
[0016] The set top terminal 400 receives entitlement information intermittently from the head end server 300, and enables a customer to access the program whose customer is a registered user between a certain time intervals (for example, accounting period). In this specification, a package is the set of a predetermined program and a certain program can belong to 1 or two or more packages. A program means all of continuous multimedia transmission of the episode of television, or specific die length like a movie. Entitlement information is downloadable in the set top terminal 400 from the head end server 300 using which suitable secure one way or bidirectional protocol.
[0017] Program key and program identifier each transmitting program is enciphered by the head end server 300 using the program key kp. This program key kp can be made unique to a program. Suitable encryption and a security technique are indicated by reference, B.Schneier, and Applied Cryptography (2d ed.1997). In addition to transmission of an encryption program, the head end server 300 also transmits a n bit program identifier to the set top terminal 400. This is used by the set top terminal 400 with the memorized entitled information, and as shown in a detail, it obtains a decode key required to decode a program in the bottom.
[0018] The program identifier p is not chosen as arbitration so that the item of the bottom entitled assignment of the program identifier to a program may explain. In a desirable example, the program identifier p can consist of the 32-bit value transmitted in the ECM field specified to MPEG-2 criterion. In this case, if it is the registered user of the program of specification [a customer], the set top terminal 400 can obtain the program key kp from the information memorized and received, and it can use the program key kp so that an encryption program may be decoded after that.
[0019] According to the further description of this invention, each of the k-bit program key kp used for an encryption transmitting program can be obtained by applying 1 or two or more pseudo-random Hash Functions to a master key m. Explanation of a suitable pseudo-random Hash Function is indicated by reference and O.Goldreich et al. and "How to Construct Random Functions" J.ACM and 33:792-807 (1986).
[0020] As an example, it is secure in cryptography, and the Hash Function which doubles die length is used as follows.
H: {0, 1} k->{0, 1}2k -- here, k is the die length of the program key kp. Therefore, Hash Function H takes the binary value of k bits, and makes the binary value of die-length 2k. The output of this Hash Function H can

be expressed as pair H0 of a k bit binary value as H1. Here, H0 is the left-hand side one half (left-hand side digit bit) of the output of Hash Function H, and is H. {1} is the right-hand side one half (right-hand side digit bit) of the output of Hash Function H. H0 and H1 can be called a separate Hash Function.

[0021] If it is k= 160, H can be specified using secret hash standard SHA-1 which is indicated by reference, Secure Hash Standard, National Institute of Standards and Technology, NIST FIPS PUB 180-1, and U.S.Dept.of Commerce (April, 1995). That is, H0 is set to SHA-1 (x||0), and H1 turns into SHA-1 (x||1). Here, 0 and 1 are the bit strings of all the bit strings 1 of 0 altogether, respectively.

[0022] The program key kp can be obtained by applying recurrently 1 or two or more Hash Functions to a master key m according to the binary value of the program identifier p. As an example, the program key kp can be obtained by applying recurrently one side of Hash Functions H0 or H1 to a master key m according to the binary value of each bit position of the program identifier p. Generally, if the program identifier p consists of n bits, according to the bit value to which the program identifier p corresponds, one side of Hash Functions H0 or H1 will be applied to each of the bit position of n of the program identifier p (it starts from a leftmost bit).

[0023] One side of Hash Functions H0 or H1 is first applied to a master key according to the binary value which is a leftmost digit bit. After that, according to the binary value which is the bit to which one side of Hash Functions H0 or H1 corresponds, it is applied to the result of pre- hash actuation to each remaining bit position (n-1). This hash actuation can be expressed as follows so that the item of a title called lower "key tree" may explain.

[Equation 2]

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

[0024] As mentioned above, the head end server 300 transmits the program identifier p with an encryption program. Therefore, if the program identifier p is given, the set top terminal 400 must obtain the program key kp used for decode of a receiving agent. As mentioned above, the program key kp can be obtained by applying recurrently 1 or two or more Hash Functions to a master key m according to the binary value of the program identifier p. The program key kp must be obtained by a customer's set top terminal 400, using indirectly the memorized entitlement information and the program identifier p which received which is explained in the bottom.

[0025] As explained on the key tree, the program key kp can be obtained by using recurrently 1 or two or more Hash Functions for a master key m according to the binary value of the program identifier p. The k-bit single master key m is used. The bit of the program identifier p can be expressed as p= (p1, ..., pn). Here, p1 is a leftmost digit bit and is a rightmost digit bit. The cryptographic key kp to the program which has the program identifier p can be defined as follows.

[Equation 3]

$$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$$

[0026] Hash actuation can be expressed as a perfect n level binary tree T like the key tree 200 shown in drawing 2 . The key tree 200 shown in drawing 2 corresponds to the example of mounting which has the program identifier p which consists of a triplet. As shown in drawing 2 , a master key m is arranged on the root 210 of a tree 200. The program key kp corresponds to a leaf node like leaf nodes 240-247. The index corresponding to each program key kp shown in drawing 2 like the index 011 corresponding to the program key kp of the DERIFU node 243 shows the pass which lets the key tree 200 from the root 210 to a leaf node 243 pass. For example, the program key kp of 243 can be obtained by following with the left edge (H0) from the root 210, the right edge (H1) from a node 220, and the right edge (H1) from a node 232. That is, H1 is further applied for H0 to the 2nd hash result. The program key kp011 can be obtained.

[0027] Therefore, the label of a node u like a node 243 is what connected the label on the edge of the pass to

Node u from the root 210. The label of each node can be specified by the program identifier p. Since the subtree which makes Node u the root is expressed, T (u) is used (namely, since the set of the program identifier p corresponding to the leaf in the subtree of Node u is expressed). The internal node u in depth r in the key tree 200 has the partial program identifier p (u1, ..., ur), and can calculate the key of which program in subtree T (u) to these. Any key of the program in the subtree of Node u is calculable by carrying out time (n-r) actuation of the Hash Function. Specifically, it uses so that the value of each bit of the low digit of (n-r) of the program identifier p may direct suitable Hash Functions H0 or H1. Therefore, the program key kp corresponding to Node u can function as an entitlement to all the programs in the subtree of Node u.

[0028] If Function H is a pseudo-random generator, mapping kp{0, 1} ->[n] {0, 1} k of the program key which the master key m parameterized is a pseudo-random function. This is indicated by reference, and O.Goldreich et al. and "How toConstruct Random Functions" J.ACM and 33:792-807 (1986).

[0029] System component drawing 3 is the block diagram showing the head end server's 300 AKI theque char. A head end shall be related with the service provider of the arbitration which transmits a television network, a cable employment person, a digital satellite service employment person, or the contents of encryption programming. the head end server 300 -- for example, IBM -- it can mount with RS6000 server which Corp(s) and manufactures, and the function and actuation of this invention can be performed. The head end server 300 is equipped with related memory like a processor 310 and the data storage device 320. A processor 310 may be mounted as a single processor and may be mounted as some processors which operate to juxtaposition. The data storage device 320 and ROM are made to memorize 1 or two or more instructions, and a processor 310 enables it to perform by taking out and interpreting.

[0030] As mentioned above, the data storage device 320 is equipped with the master key database 350 which memorizes a master key m. For example, a master key m can be updated like [for every accounting period]. Moreover, the data storage device 320 has the program database 500 so that it may explain in relation to drawing 5 in the bottom. The program database 500 presents the program identifier p and the related package corresponding to each program. moreover, drawing 7 R> -- the data storage device 320 has the entitlement information delivery process 700 and the program delivery process 800 so that it may explain in relation to 7 and 8.

[0031] Generally, the entitlement information delivery process 700 generates and distributes the entitlement information which each customer needs to accessing the program which is a registered user. Moreover, the program delivery process 800 obtains the program key kp based on the program identifier p assigned to the program, in order to encipher a program and to transmit by the program identifier p.

[0032] The communication link port 330 links the head end server 300 to each connected receiver like the set top terminal 400 which showed the head end server 300 to the network 110 at a bond and drawing 1 .

[0033] Drawing 4 is the block diagram showing the AKI theque char of the set top terminal 400. The set top terminal 400 can be mounted as a set top terminal (STT) corresponding to television, and it can be changed so that the function and actuation of this invention may be performed. The set top terminal 400 is equipped with a processor 410 and memory like data storage 420, and the communication link port 430, and operates by the same approach as the above hardware relevant to drawing 3 .

[0034] Data storage 420 is equipped with the entitlement database 600 memorizable into the secure part of data storage 420 so that it may explain in relation to drawing 6 in the bottom. The entitlement database 600 contains the part of the key tree 200 required in order that a customer may get the program key kp to the program which has an entitlement. Moreover, data storage 420 is equipped with Hash Functions H0 and H1 (440). Moreover, data storage 420 includes the decoding process 900 so that it may explain in relation to drawing 9 in the bottom. Generally, using the program identifier p received in order to obtain the program key kp, and the memorized entitlement information 600, in order to decode a program, the program key kp is used for the decoding process 900, and it decodes the program whose customer has an entitlement.

[0035] Drawing 5 shows the program database 500 which memorizes information on each program p transmitted by the head end server 300. This information is transmitted to for example, an accounting period with the program identifier p to which that program belongs and which packs and corresponds. The program database 500 holds two or more decodings like records 505-520. These are related with a different program, respectively. The program database 500 contains the program identifier p which corresponds in the field 535 including directions of the corresponding package with which the program belongs in the field 530 to each program identifier identified by the program name in the field 525.
[0036] Drawing 6 shows the entitlement database 600 containing the part of the key tree 200 required for a customer to get the program key kp to the program which has an entitlement. As mentioned above, T (u) expresses the set of the program identifier p corresponding to the leaf nodes 240-247 in the subtree which makes Node u the root, i.e., the subtree of Node u. For example, supposing a customer has an entitlement about receiving four programs corresponding to leaf nodes 240-243, entitlement information will consist of a middle key corresponding to a node 220. In this approach, if needed, suitable Hash Functions H0 and H1 (440) can be used in order to obtain the program key kp to each nodes 230, 232, 240-243 in the subtree of a node 220.
[0037] The entitlement database 600 shown by drawing 6 is a registered user who receives four programs corresponding to leaf nodes 240-243 (there is an entitlement), and is a registered user who receives two programs corresponding to leaf nodes 246-247. Therefore, the entitlement information recorded on the entitlement database 600 consists of a middle key corresponding to a node 220 and a node 236. nodes 220 and 236 -- it is alike, respectively, and it receives, and the entitlement information recorded on the entitlement database 600 has the middle key values kio and ki11, respectively, and has corresponding directions of the partial program identifier p. The approach by which the entitlement database 600 is generated by the entitlement information delivery process 700 based on the package of the program which the customer chose is explained in relation to drawing 7 in the bottom.
[0038] A small entitlement is establishable to the set of many programs of various sizes using the tree method of program packaging this invention. The target set S is established using the set of the program packed. The minimum set of a tree node with which a subtree covers the target set S correctly is obtained as follows.
[Equation 4]
$T(S)=Z\subseteq T$ ただし、$\bigcup_{u\in Z} T(u)=S$、かつ、$|Z|$ は最小であるように

[0039] The entitlement information over Package S is the set ki of the middle key held in the node of T (S). As shown in a top, the set top terminal 400 decodes the program in S (accepting it) correctly with the set of this key. Theoretically, the tree method of this invention can build the entitlement information over the target set S of which arbitration. furthermore -- however, if the program identifier p is assigned to arbitration, entitlement information will become so large that it is not allowed for the secure memory to which the set top terminal 400 was restricted.
[0040] a process -- as mentioned above, the head end server 300 performs the entitlement information delivery process 700 shown in drawing 7 , and generates and distributes the entitlement database 600 required for each user in order to access the program which is a registered user. As mentioned above, the entitlement database 600 consists of corresponding directions and the corresponding middle key value ki of a partial program identifier to each node of the key tree 200 required for a customer to get the program key kp to the program which is a registered user.
[0041] Therefore, the entitlement information delivery process 700 identifies first the program which the customer chose (710). After that, the entitlement information delivery process 700 finds minimum set [of a tree node] T (S). The subtree covers the target set S correctly. The target set S is disassembled to the

maximum De Dis joint interval of the KONSEKYUTIBU program identifier p (720). Two program identifiers p are considered to be KONSEKYUTIBU when the integer over the binary expression is KONSEKYUTIBU.
[0042] And covering T (S) is found to each interval (730). The corresponding partial program identifier p held in the node of covering T (S) to Set ki and each interval of a middle key is generated (740). At the end, the generated entitlement information downloads to the set top terminal 400 with the head end server 300 (750), and program control is completed (760).
[0043] The number of the intervals in the target set S can be set to I (S). In order to calculate covering T (S) to the single interval of the program identifier p to the order of the tree node of n, the key tree 200 of depth n must be asked. Therefore, the time amount complexity of the entitlement information delivery process 700 serves as order of I(S) -n. Similarly, the magnitude of minimum covering T (S) serves as order of I(S) -n. The program identifier p which enables the program of related contents to carry out packaging of them efficiently should be assigned. In an example, a fundamental package is the gestalt of all the program identifiers p that have the bit prefix mu.
[0044] The entitlement of such a single topic package is a single key in the key tree 200. Moreover, a multi-topic package can be assembled without a side effect. Entitlement information is only the set of a key to each TOPICS which consists of a multi-TOPICS package. According to this invention, the package specified by Prefix mu does not force to the set top terminal 400 so that a program may be decoded using zero prefix of the same die length.
[0045] As mentioned above, the head end server 300 performs the program delivery process 800 shown in drawing 8 , and in order to decode a program and to transmit using the program identifier p, he gets the program key kp based on the program identifier p assigned to the program and the master key m. The program delivery process 800 is important for performing in off-line thru/or the real time except an actual transmitting step. As shown in drawing 8 , the program delivery process 800 starts the process using the principle of this invention by identifying the program which should be transmitted (810).
[0046] After that, the program delivery process 800 takes out the program identifier p corresponding to the program from the program database 500 (820), and calculates the program key kp corresponding to the program (830). And a program is enciphered using the program key kp calculated at the front step (840). Finally, the program delivery process 800 transmits the program enciphered with the program identifier p (850), and program control ends it (860).
[0047] It is important to suppose that it is possible to obtain the program key kp required for the program identifier p to be interleaved periodically, able to transmit it through transmission of program information, and for a customer change a channel at the time of a program, and decode a program. In another example, the program identifier p can be continuously transmitted on another control channel like a Barker channel.
[0048] As mentioned above, the set top terminal 400 performs the decoding process 900 shown in drawing 9 , using the entitlement information 600 and the received program identifier p memorized in order to obtain the program key kp, in order to decode the program, the program key kp is used and a customer decodes the program by which the entitlement is carried out. As shown in drawing 9 , the decoding process 900 starts the process which used the principle of this invention on the occasion of the reception of the customer directions made to tune up to a specific channel (910).
[0049] After that, the set top terminal 400 receives the suitable signal containing the enciphered program identifier p which was programmed and transmitted (920). The decoding process 900 takes out the entitlement information memorized from the entitlement database 600 (930). It judges whether the transmitted program is included (940). When the entry which has the partial-program identifier p which agrees in the leftmost digit bit of the receiving-agent identifier p at step 940 is judged not to exist in the entitlement database 600, a customer does not have an entitlement to the selected program and program control is ended (980).
[0050] However, if an entry exists in the entitlement database 600 which has the partial-program identifier p

corresponding to the leftmost digit bit of the received program identifier p, a customer has an entitlement to the selected program. Therefore, the program key kp is calculated using the middle key ki taken out from the entry of the entitlement database 600 (960). Specifically, the program key kp is calculated by operating suitable Hash Functions H0 or H1 so that each value of the bit of the low (n-r) order of the program identifier p may direct as follows.

[Equation 5]

$$K_p = H_{p_n}(\ldots H_{p_{r+1}}(H_{p_r}(K_I))\ldots)$$

[0051] Finally, the program is decoded using the obtained program key kp (970), and ends program control (980). When the received program is not a part of a customer's entitlement here, it is important that there is no entitlement information which has the partial identifier p corresponding to the low bit of the program identifier p which received with the transmitting program in the entitlement database 600.

[0052] The decoding process 900 obtains a decode key, or moreover, as mentioned above Before a customer judges whether there is any entitlement to a demand channel In order that it can wait for a customer to demand a specific channel and the decoding process 900 may obtain the transmitting program identifier p instead, all channels are scanned periodically. It is important that the decode key to the storage in data storage 420 can be obtained, and a customer's entitlement can be judged beforehand again.

[0053] a suitable Hash Function -- as mentioned above, if Hash Function H is a pseudo-random bit generation machine, it can prove that mapping of p->kp is a pseudo-random function. Therefore, a code key cannot be predicted if actual Hash Function H is strong in cryptography. Therefore, if a piracy person has access only to encryption program broadcasting, it will not be able to break through a code in the knowledge about the key generated using the tree method of this invention. Therefore, only one concerns only become ensuring that video encryption algorithm can oppose to a well-known plain text attack.

[0054] Hash Function H should hold two properties. Calculating Input x has that it must be difficult noting that the one half H0 of an image (x) or H1 (x) is given to the 1st to Hash Function H. Though this knows the image of both these one half, it is actually materialized also to the cryptography-hash [which] H with it difficult [to carry out an inverted arch]. Moreover, though H1 (x) was known, it must be difficult to calculate H0 (x), and the reverse of a thing is also the same. Even if it is difficult fundamentally to carry out the inverted arch of the function H, when the key of one one half is known, it becomes easier to complete the key of the remaining one half. If that is right, the piracy person who knows Program kp to Node u can calculate the key to the SHIBURINGU (sibling: sibling) node v, and can calculate the key to all the programs in the subtree of Node v.

[0055] As one advantage of the tree method according to this invention, merge of an entitlement carried out in piracy may be made in inefficient. Pair p1, p2, and those ********** of a SHIBURINGU program are considered. A piracy person assumes that the program key kp corresponding to the programs p1 and p2 of both which are two one half of H (kp (u)) is known. A piracy person still cannot do the inverted arch of the H, and cannot calculate kp (u). It is because H is a cryptography-Hash Function. Therefore, the entitlement carried out in the merged piracy must contain both kp (p1) and kp (p2) instead of compact kp (u). therefore, it is not a strategy good for a piracy person to divide to two or more set top terminals 400 which use a CHIPU (it is -- although -- it differs) entitlement. It is because a union ****** entitlement becomes very large.

[0056] As mentioned above, the suitable pseudo-random Hash Function is indicated by reference, and O.Goldreich et al. and "How to Construct Random Functions" J.ACM and 33:792-807 (1986).

[Translation done.]

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] The block diagram showing the system which transmits the enciphered contents of programming according to one example of this invention.
[Drawing 2] Drawing showing the example of the key tree according to this invention.
[Drawing 3] The block diagram of the head end server of drawing 1 .
[Drawing 4] The block diagram of the set top terminal of drawing 1 .
[Drawing 5] The table from the program database of drawing 3 .
[Drawing 6] The table from the entitled database of drawing 4 .
[Drawing 7] The flow chart showing the entitlement information delivery process which the head end server of drawing 3 uses.
[Drawing 8] The block diagram showing the program distribution flow chart which the head end server of drawing 3 uses.
[Drawing 9] The flow chart showing the record process which the set top terminal of drawing 4 uses.
[Description of Notations]
110 Distribution Network
200 Key Tree
220, 230, 232, 236, 240-243, 246-247 Node
300 Head End Server
310 410 Processor
320 420 Data storage
350 databases
330 430 Communication link port
400-401 Set top terminal
440 Hash Functions H0 and H1
500 Program Database
505-520 Decoding
525, 530, 535 Field
600 Entitlement Database
700 Entitlement Information Delivery Process
710 Identify Program Which Customer Chose.
720 Decompose to the Maximum De Dis Joint Interval of Target Set KONSEKYUTIBU Program Identifier P.
730 Find Covering T (S) to Each Interval.
740 Generate Partial-Program Identifier P to which Middle Key Ki Sets and Corresponds in Node of Covering T (S) to Each Interval.
750 Transmit Entitlement Information to Set Top Terminal.
760, 860, 980 Termination
800 Program Delivery Process

810 Identify Program Which Should be Transmitted.
820 Take Out Program Identifier P from Program Database.
830 Calculate Program Key.
840 Encipher Program Using Program Key.
850 Transmit Program Enciphered with Program Identifier P.
900 Decoding Process
910 Take Out Customer Directions Made to Tune Up to Channel.
920 Receive Sending Signal Containing Program and Program Identifier P.
930 Take Out Entitlement Information Memorized from Entitlement Database.
940 Is There an Entry Which Has Partial-Program Identifier P corresponding to MSB of Receiving-Agent Identifier P?
960 Come Out Picking and Calculate Program Key Kp Using Ki Value and Hash Functions H0 and H1 Bottom.
970 Decode Program Using Program Key Kp.

[Translation done.]

DESCRIPTION OF DRAWINGS

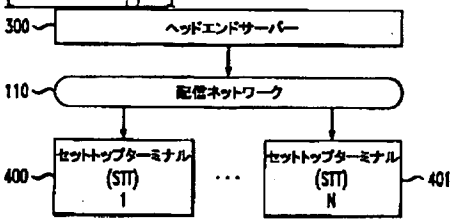
[Brief Description of the Drawings]
[Drawing 1] The block diagram showing the system which transmits the enciphered contents of programming according to one example of this invention.
[Drawing 2] Drawing showing the example of the key tree according to this invention.
[Drawing 3] The block diagram of the head end server of drawing 1 .
[Drawing 4] The block diagram of the set top terminal of drawing 1 .
[Drawing 5] The table from the program database of drawing 3 .
[Drawing 6] The table from the entitled database of drawing 4 .
[Drawing 7] The flow chart showing the entitlement information delivery process which the head end server of drawing 3 uses.
[Drawing 8] The block diagram showing the program distribution flow chart which the head end server of drawing 3 uses.
[Drawing 9] The flow chart showing the record process which the set top terminal of drawing 4 uses.
[Description of Notations]
110 Distribution Network
200 Key Tree
220, 230, 232, 236, 240-243, 246-247 Node
300 Head End Server
310 410 Processor
320 420 Data storage
350 databases
330 430 Communication link port
400-401 Set top terminal
440 Hash Functions H0 and H1
500 Program Database
505-520 Decoding
525, 530, 535 Field
600 Entitlement Database
700 Entitlement Information Delivery Process
710 Identify Program Which Customer Chose.
720 Decompose to the Maximum De Dis Joint Interval of Target Set KONSEKYUTIBU Program Identifier P.
730 Find Covering T (S) to Each Interval.
740 Generate Partial-Program Identifier P to which Middle Key Ki Sets and Corresponds in Node of Covering T (S) to Each Interval.
750 Transmit Entitlement Information to Set Top Terminal.
760, 860, 980 Termination
800 Program Delivery Process

810 Identify Program Which Should be Transmitted.
820 Take Out Program Identifier P from Program Database.
830 Calculate Program Key.
840 Encipher Program Using Program Key.
850 Transmit Program Enciphered with Program Identifier P.
900 Decoding Process
910 Take Out Customer Directions Made to Tune Up to Channel.
920 Receive Sending Signal Containing Program and Program Identifier P.
930 Take Out Entitlement Information Memorized from Entitlement Database.
940 Is There an Entry Which Has Partial-Program Identifier P corresponding to MSB of Receiving-Agent Identifier P?
960 Come Out Picking and Calculate Program Key Kp Using Ki Value and Hash Functions H0 and H1 Bottom.
970 Decode Program Using Program Key Kp.

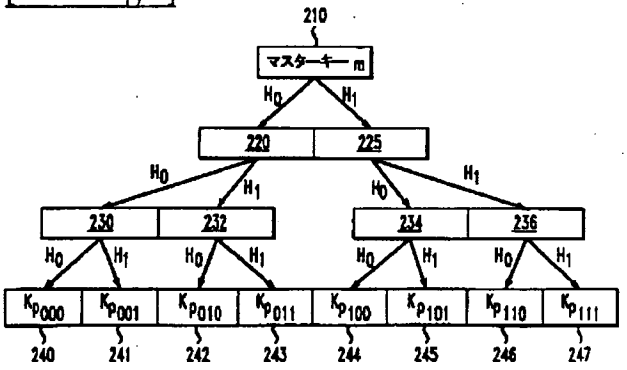
[Translation done.]

DRAWINGS

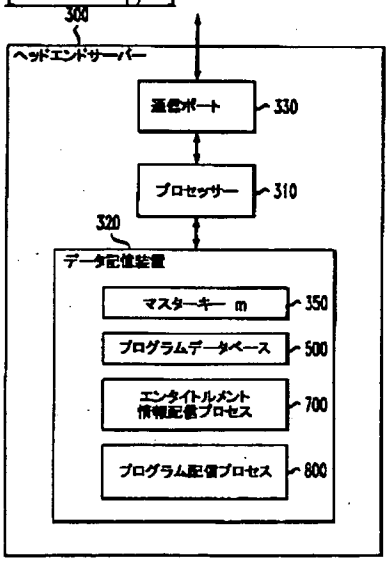
[Drawing 1]



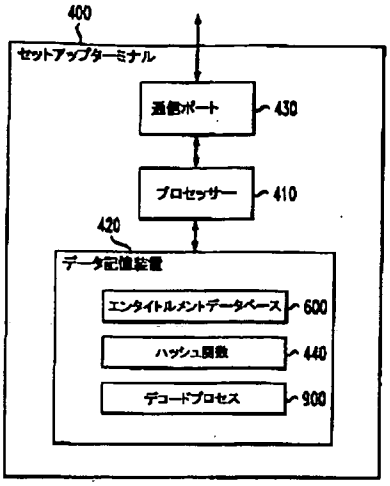
[Drawing 2]



[Drawing 3]



[Drawing 4]



[Drawing 5]

プログラムデータベース 500

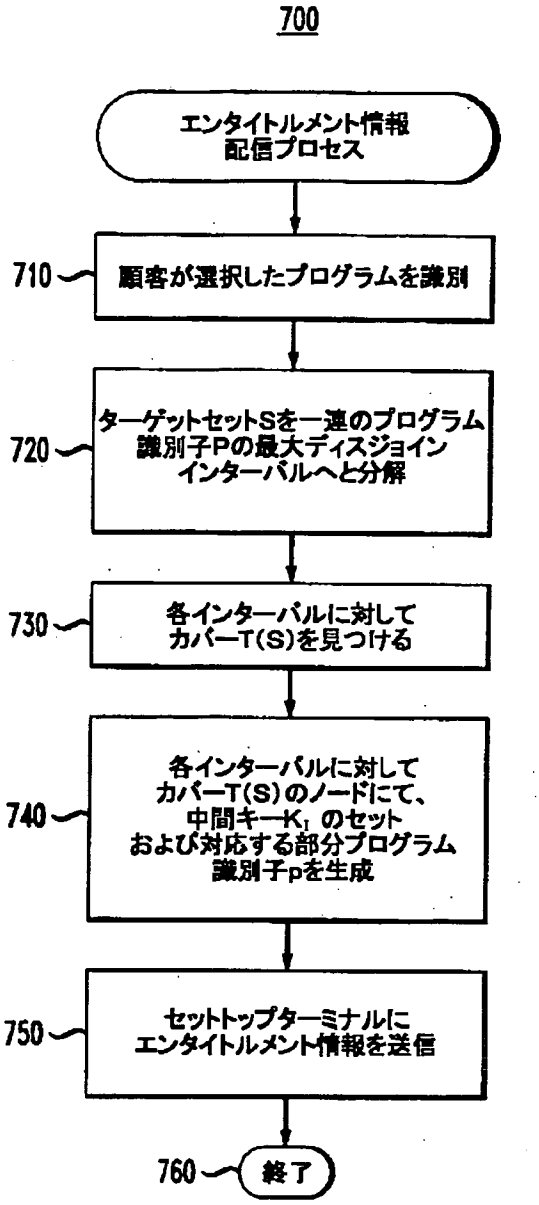
	525 プログラム	530 パッケージ名	535 プログラム識別子
505	ワールドシリーズ試合5	スポーツ、プロ野球、プレーオフ試合	p^1
510	スーパーボール	スポーツ、プロフットボール、プレーオフ試合	p^2
515	サウンドオブミュージック	映画、ミュージカル	p^3
520	セサミストリート エピソード第554	子供向けプログラム 教育用プログラム	p^4

[Drawing 6]

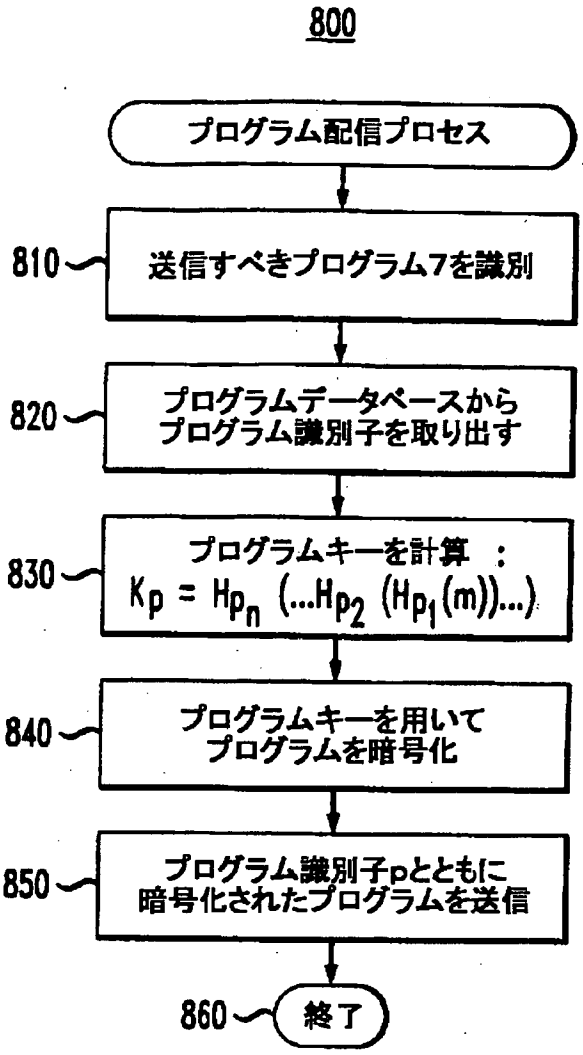
エンタイトルメントデータベース 600

ノード	キー値	部分プログラム識別子
220	KI_0	0
236	KI_{11}	11

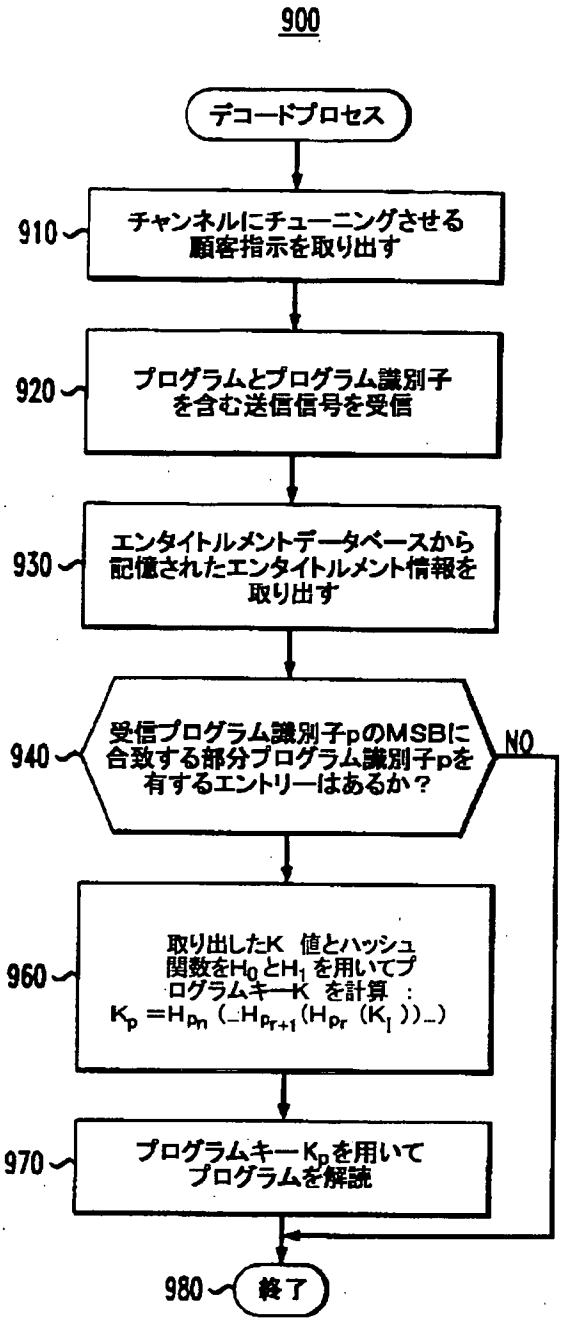
[Drawing 7]


700
エンタイトルメント情報
配信プロセス
710
顧客が選択したプログラムを識別
720
ターゲットセットSを一連のプログラム
識別子Pの最大ディスジョイン
インターバルへと分解
730
各インターバルに対して
カバーT(S)を見つける
740
各インターバルに対して
カバーT(S)のノードにて、
中間キーK_i のセット
および対応する部分プログラム
識別子pを生成
750
セットトップターミナルに
エンタイトルメント情報を送信
760
終了

[Drawing 8]


800
プログラム配信プロセス
810
送信すべきプログラム7を識別
820
プログラムデータベースから
プログラム識別子を取り出す
830
プログラムキーを計算 :
$K_p = H_{p_n}(\ldots H_{p_2}(H_{p_1}(m))\ldots)$
840
プログラムキーを用いて
プログラムを暗号化
850
プログラム識別子pとともに
暗号化されたプログラムを送信
860
終了

[Drawing 9]



[Translation done.]

CORRECTION OR AMENDMENT

[Kind of official gazette] Printing of amendment by the convention of 2 of Article 17 of Patent Law
[Section partition] The 3rd partition of the 7th section
[Publication date] November 8, Heisei 14 (2002. 11.8)

[Publication No.] JP,2001-36517,A (P2001-36517A)
[Date of Publication] February 9, Heisei 13 (2001. 2.9)
[Annual volume number] Open patent official report 13-366
[Application number] Application for patent 2000-135069 (P2000-135069)
[The 7th edition of International Patent Classification]

H04L 9/08
G09C 1/00 650
H04N 5/44
7/08
7/081
7/16
7/167

[FI]

H04L 9/00 601 D
G09C 1/00 650 Z
H04N 5/44 A
7/16 C
H04L 9/00 601 E
H04N 7/08 Z
7/167 Z

[Procedure revision]
[Filing Date] August 13, Heisei 14 (2002. 8.13)
[Procedure amendment 1]
[Document to be Amended] Specification
[Item(s) to be Amended] Claim
[Method of Amendment] Modification
[Proposed Amendment]
[Claim(s)]
[Claim 1] It is the approach of transmitting the program which can carry out access restriction to an end user,
(A) The step which assigns the program identifier which has a binary value to said program,
(B) The step which defines at least one master key,

(C) The step which enciphers said program by using the program key obtained by applying at least one Hash Function to said master key based on the binary value of said program identifier,
(D) The approach characterized by having the step which sends said enciphered program to said end user with said program identifier.
[Claim 2] Said program identifier is an approach according to claim 1 characterized by applying one of said the Hash Functions to each location of n bits of said program identifier according to the bit value to which it becomes from n bits and said program identifier corresponds.
[Claim 3] (E) The approach according to claim 1 characterized by having further the step which provides said end user with entitlement information based on the set of the program acquired by said end user.
[Claim 4] The approach according to claim 3 characterized by including some key trees based on the set of the program acquired by said end user in said entitlement information.
[Claim 5] Said end user is an approach according to claim 3 characterized by using said program identifier in order to obtain said program key from said memorized entitlement information.
[Claim 6] Said program identifier is an approach according to claim 1 characterized by interleaving with transmission of said encryption program.
[Claim 7] Said program identifier is an approach according to claim 1 characterized by being transmitted on a control channel.
[Claim 8] It is the approach of transmitting a program to two or more end users,
(A) The step enciphered using the program key obtained by applying a Hash Function to the master key based on the binary value of each bit position of said program identifier for the program which has a program identifier recurrently,
(B) The approach characterized by having the step which transmits the enciphered program and said program identifier to said end user.
[Claim 9] It is the approach of transmitting the program corresponding to at least one program package to two or more end users,
(A) The step which provides said end user with entitlement information based on the set of the program acquired by said end user,
(B) The step enciphered using the program key obtained by applying a Hash Function to the master key based on the binary value of each bit position of said program identifier for the program which has a program identifier recurrently,
(C) It has further the step which transmits said program identifier to said end user with the enciphered program,
It is the approach characterized by obtaining said program key from the entitlement information said end user was remembered to be when said end user was a just user of said program.
[Claim 10] It is the approach of decoding the enciphered program,
(A) The step which receives the entitlement information which contains at least one middle key from a key tree based on the set of the program which said customer acquired from the provider of said program,
(B) The encryption program enciphered by the program key, and the step which receives a program identifier,
(C) The step which obtains said program key from the part said program identifier and said key tree were remembered to be,
(D) The approach characterized by having the step which decodes said encryption program using said program key.
[Claim 11] Said program identifier consists of n bits,
It is the approach according to claim 10 which said master key is arranged on the root of said key tree, and is characterized by generating said key tree when said key tree applies a Hash Function to each node until the tree level of n is made.

[Claim 12] It is the approach of decoding the enciphered program,
(A) The step which receives the entitlement information which contains at least one middle key from the key tree based on the set of the program which a customer acquires from the provider of said program,
(B) The encryption program enciphered by the program key, and the step which receives a program identifier,
(C) The step which obtains said program key from the part the key tree was remembered to be from said program identifier and said middle key by applying a Hash Function to said middle key recurrently based on the binary value of said program identifier,
(D) The approach characterized by having the step which decodes said encryption program using said program key.
[Claim 13] Said program identifier consists of n bits,
It is the approach according to claim 12 which said middle key corresponds to the intermediate node in the level r of said key tree, and is characterized by carrying out n-r time application of said Hash Function at said middle key.
[Claim 14] It is the system which transmits the program which restricts access to an end user,
(A) Memory which memorizes a master key and a computer readout possible code,
(B) It has the processor connected with said memory in actuation, and this processor,
(a) Assign the program identifier which has a binary value to said program,
(b) Define at least one master key,
(c) Encipher said program using a program key by applying at least one Hash Function to said master key based on the binary value of said program identifier,
(d) The system characterized by constituting so that an encryption program may be transmitted to said end user with said program identifier.
[Claim 15] It is the system which transmits the program to which access to an end user was restricted,
(A) Memory which memorizes a master key and the code which can be computer read,
(B) It has the processor connected with said memory on actuation,
Said processor,
(a) Encipher this program that has a program identifier using the program key obtained by applying a Hash Function to a master key recurrently based on the binary value of each bit position of said program identifier,
(b) The system characterized by constituting so that this program enciphered by said end user and said program identifier may be transmitted.
[Claim 16] It is the system which decodes the enciphered program,
(A) Memory which memorizes a master key and the code which can be computer read,
(B) It has the processor connected with said memory on actuation, and is said processor,
(a) Receive the entitlement information containing the part of the key tree based on the set of the program acquired by said customer from the provider of this program,
(b) Receive the encryption program enciphered by the program key and a program identifier,
(c) Obtain said program key from said part said program identifier and said key tree were remembered to be,
(d) The system characterized by constituting so that said encryption program may be decoded using said program key.
[Claim 17] It is the medium by which the code means which can be computer read was mounted and which can be computer read, and this means that can be computer read is at the time of operation,
(a) Assign the program identifier which has a binary value to a program,
(b) Define at least one master key,
(c) Encipher this program using the program key obtained by applying at least one Hash Function to said

master key based on the binary value of said program identifier,
(d) The medium which is characterized by transmitting this program enciphered with said program identifier to an end user and which can be computer read.
[Claim 18] It is the medium by which the code means which can be computer read was mounted and which can be computer read, and this means that can be computer read is at the time of operation,
(a) Receive the entitlement information containing the part of the key tree based on the set of the program acquired by said customer from the provider of this program,
(b) Receive the encryption program enciphered by the program key and a program identifier,
(c) Obtain said program key from said part said program identifier and said key tree were remembered to be,
(d) The medium which is characterized by decoding said encryption program using said program key and which can be computer read.

[Translation done.]